The Double Hierarchy Method. A parallel 3D contact method for the interaction of spherical particles with rigid FE boundaries using the DEM

Miquel Santasusana, Joaquín Irazábal, Eugenio Oñate & Josep Maria Carbonell

Computational Particle Mechanics

ISSN 2196-4378

Comp. Part. Mech.
DOI 10.1007/s40571-016-0109-4

Springer

Comp. Part. Mech.
DOI 10.1007/s40571-016-0109-4

The Double Hierarchy Method. A parallel 3D contact method for the interaction of spherical particles with rigid FE boundaries using the DEM

Miquel Santasusana[1] · **Joaquín Irazábal**[1] · **Eugenio Oñate**[1] · **Josep Maria Carbonell**[1]

Received: 3 February 2016 / Revised: 16 March 2016 / Accepted: 17 March 2016

Abstract In this work, we present a new methodology for the treatment of the contact interaction between rigid boundaries and spherical discrete elements (DE). Rigid body parts are present in most of large-scale simulations. The surfaces of the rigid parts are commonly meshed with a finite element-like (FE) discretization. The contact detection and calculation between those DE and the discretized boundaries is not straightforward and has been addressed by different approaches. The algorithm presented in this paper considers the contact of the DEs with the geometric primitives of a FE mesh, i.e. facet, edge or vertex. To do so, the original hierarchical method presented by Horner et al. (J Eng Mech 127(10):1027–1032, 2001) is extended with a new insight leading to a robust, fast and accurate 3D contact algorithm which is fully parallelizable. The implementation of the method has been developed in order to deal ideally with triangles and quadrilaterals. If the boundaries are discretized with another type of geometries, the method can be easily extended to higher order planar convex polyhedra. A detailed description of the procedure followed to treat a wide range of cases is presented. The description of the developed algorithm and its validation is verified with several practical examples. The parallelization capabilities and the obtained performance are presented with the study of an industrial application example.

Keywords Discrete element method · Contact detection · Hierarchical search · Particle–solid contact interaction

✉ Joaquín Irazábal
jirazabal@cimne.upc.edu

Miquel Santasusana
msantasusana@cimne.upc.edu

[1] International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain

1 Introduction

Since Cundall [7] presented the first ideas about the DEM in 1979, this numerical technique has increased its popularity, being, nowadays, one of the most powerful and efficient tools to reproduce the behaviour of granular materials. Within the DEM approach, each material grain is represented as a rigid particle. The deformation of the material is represented by the interaction between the particles. As the particles are rigid, the normal and tangential contact between them define the material constitutive behaviour.

Due to the method formulation, the definition of the appropriate contact laws is fundamental and a fast contact detection is something of significant importance in DEM calculations. Contact status between individual objects, which can be two DE particles (DE/DE contact) or a DE particle and a boundary element (e.g. DE/FE contact) can be calculated from their relative position in the previous time step and it is used for updating the contact forces at the current step. The relative computational cost of the contact detection over the total computational cost is high in most of DEM simulations, and so, the problem of how to recognize all contacts precisely and efficiently has received considerable attention in the literature [26,41]. Note that the term FE used in this article does not refer to the solution of any partial equation by means of the finite element method. Instead the term is used to refer to the geometry elements (triangles, quadrilaterals, etc.) which are used to discretize the boundaries.

Traditionally, the contact detection is split into two stages: Global Neighbour Search and Local Contact Resolution. Concerning Global Neighbour Search the computational cost can be reduced from $\mathcal{O}(N^2)$ in an all-to-all check to a $\mathcal{O}(N \cdot ln(N))$. Han et al. [16] compared the most common Global Neighbour Search algorithms (cell-based and tree-based) in simulations with spherical particles. Numerical

Published online: 01 April 2016

tests showed better performance for the cell-based algorithms (D-Cell [42] and NBS [27]) over the tree-based ones (ADT [2] and SDT [12]), specially for large-scale problems. It should be noted also that the efficiency is dependent on the cell dimension and, in general, the size distribution can affect the performance. Han et al. [16] suggest a cell size of three times the average discrete object size for 2D and five times for 3D problems. It is worth noting that using these or other efficient algorithms, the cost of the Global Neighbour Search represents typically less than 5 % of the total computation, while the total cost of the search can reach values over 75 % [17], specially when the search involves non-spherical geometries such as triangles, quadrilaterals, etc. In this sense the focus should be placed on the Local Contact Resolution check rather than on optimizing the Global Neighbour Search algorithms.

Regarding the characterization of the contact boundaries, several solutions have been reported for resolving particle–solid interaction problems. Among the simplest ones is the glued-sphere approach [21], which approximates any complex geometry (i.e. a rigid body or boundary surface) by a collection of spherical particles so it retains the simplicity of particle-to-particle contact interaction. This approach, however, is geometrically inaccurate and computationally intensive due to the introduction of an excessive number of particles. Another easy approach (used in some numerical codes, e.g. ABAQUS) is to define the boundaries as analytical surfaces. This approach is computationally inexpensive, but it can only be applied in certain specific scenarios, where the use of infinite surfaces does not disturb the calculation. A more complex approach which combines accuracy and versatility is to resolve the contact of particles (spheres typically) with a finite element boundary mesh. These methods take into account the possibility of contact with the primitives of the FE mesh surface, i.e. facet, edge or vertex contact.

Horner et al. [17] and Kremmer et al. [22] developed the first hierarchical contact resolution algorithms for contact problems between spherical particles and triangular elements, while Zang et al. [44] proposed similar approaches accounting for quadrilateral facets. Dang et al. [9] upgraded the method introducing a numerical correction to improve smoothness and stability. Su et al. [34] developed a complex algorithm involving polygonal facets under the name of RIGID method which includes an elimination procedure to resolve the contact in different non-smooth contact situations. This approach, however, does not consider the cases when a spherical particle might be in contact with the entities of different surfaces at the same time (multiple contacts) leading to an inaccurate contact interaction. The upgraded RIGID-II method presented later by Su et al. [35] and also the method proposed by Hu et al. [18] account for the multiple contact situations, but they have a complex elimination procedure with many different contact scenarios to distinguish, which is difficult to code in practice. Recently Chen et al. [6] presented a very simple and accurate algorithm which covers many situations. Their elimination procedure, however, requires a special database which cannot be computed in parallel.

In this work, the Double Hierarchy Method (H^2), for Local Contact Resolution, is introduced. It consists in a simple contact algorithm based not only on the FE boundary approach which has been specially designed to resolve efficiently the intersection of spheres with triangles and planar quadrilaterals but it can also work fine with any other higher order planar convex polyhedra. A two-layer hierarchy is applied upgrading the classical hierarchy method presented by Horner [17]; namely hierarchy on contact type followed by hierarchy on distance. The first one classifies the type of contact (facet, edge or vertex) for every contacting neighbour in a hierarchical way, while the distance-based hierarchy determines which of the contacts found are valid or relevant and which ones have to be removed.

Another important issue considered in this work is the parallel computation which is crucial for practical purposes of a DEM code. Industrial applications may involve a large number of particles and also a fine definition of the boundaries which, in our case, turns into large number of FE boundary conditions. Therefore, the algorithm presented here has been designed in a way that the code can work efficiently in parallel computations. This is a clear advantage over the above-mentioned publications. Nakashima [28] whose method is presumably parallelizable and Zang [44] and Su [35] which remark the importance of the future parallelization of their algorithms are exceptions.

Summarizing, the contact search framework presented is designed to satisfy the following requirements:

- Include poly-disperse elements for both: FEs and DEs.
- Allow different FE geometries and primitives (triangle, quadrilateral, polygon).
- Ensure contact continuity in non-smooth regions (edges and vertices).
- Resolve multiple contacts and contact with different entities simultaneously.
- Need low memory storage.
- Be simple, fast and accurate.
- Be fully parallelizable.

Table 1 summarizes the strengths and drawbacks of the reviewed contact detection methods. Methods based in storing all the potential contacts to lately remove the invalid ones (RIGID-II [35], Hu et al. [18], Chen et al. [6] and H^2) are the most accurate. They treat the cases with large indentations (relative to the size of the FE) and give a solution for the contact continuity in non-smooth regions of the boundary. These methods have, however, some limitations due to

Table 1 Strengths and drawbacks of the different contact detection algorithms evaluated

	Glued sph. [21]	Analytical	Hierarchical [9,17,22,44]	RIGID [34]	RIGID-II [35]	Hu [18]	Chen [6]	H^2
Wide size rate DEs/FEs	–	–	×	✓	✓	✓	×	✓
Contact element typologies	×	–	✓	✓	✓	×	×	✓
Boundary shape flexibility	✓	×	✓	✓	✓	✓	✓	✓
Multi-contact	✓	–	✓	×	✓	✓	✓	✓
Simple	✓	✓	✓	×	×	×	✓	✓
Efficient	×	✓	×	✓	×	×	✓	✓
Accurate	×	×	✓	×	✓	✓	×	✓
Low storage	✓	✓	×	×	×	×	✓	✓
Upgradable to deformable FEs	×	×	✓	✓	✓	✓	✓	✓
Large indentation	×	✓	×	×	✓*	✓*	×	✓*
Contact continuity	×	-	✓*	×	✓*	✓*	✓*	✓*

Symbol (✓) implies that the method satisfies the property. On the other hand, symbol (×) means that the method does not satisfy the property. Symbol (–) denotes that the property does not apply to that method and (✓*) means that, although method satisfies the property, there are some limitations

the fact that the real deformed geometry of the sphere is not represented in the DEM. The indentation represents, instead, the local deformation near the contact region. Due to this fact, in concave transitions of the boundary, error in the contact detection is common for all these methods including the H^2 method here presented. In Sect. 7 this is analysed under the H^2 framework and a bound of the error is provided for different situations.

The paper starts with the introduction of the basic formulation of the DE method and the classical algorithms for DE/DE contact search. Next the proposed strategy for the DE/FE contact search which includes the novel Double Hierarchy Method (H^2) is described. Some validation analysis together with examples of performance in critical situations (where most of the literature methods would fail) is presented. Finally some results of scalability and parallel computation are given.

2 DEM formulation

The algorithm has been implemented in the DEM code called DEMPack (www.cimne.com/dempack/) which is inside *Kratos* [8], an Open-Source software framework for the development of numerical methods for solving multidisciplinary engineering problems.

2.1 Basic DE formulation

The DEM is based on the characterization of the material by means of defining the interactions between constituent particles. The interaction is described by the DE/DE contact mechanics defined by the particle kinematics. In the basic DEM formulation, the translational and rotational motion of particles is defined by the standard equations for the dynamics of rigid bodies. For every spherical particle, these equations can be written as

$$m\,\ddot{\mathbf{u}} = \mathbf{F}\,, \tag{1}$$

$$\mathbf{I}\,\dot{\boldsymbol{\omega}} = \mathbf{T}\,, \tag{2}$$

where $\mathbf{u}$, $\dot{\mathbf{u}}\ddot{\mathbf{u}}$ are, respectively, the particle centroid displacement, its first and second derivative in a fixed coordinate system $\mathbf{X}$, $\boldsymbol{\omega}$ the angular velocity, $\dot{\boldsymbol{\omega}}$ the angular acceleration, m the particle mass, $\mathbf{I}$ the inertia dyadic with respect to particle centre of mass, $\mathbf{F}$ the resultant force and $\mathbf{T}$ the resultant moment about the central axes.

$\mathbf{F}$ and $\mathbf{T}$ are computed as the sum of (i) all forces and moments applied to the particle due to external loads, $\mathbf{F}^{\text{ext}}$ and $\mathbf{T}^{\text{ext}}$, respectively, (ii) the contact interactions with neighbouring spheres and boundary finite elements $\mathbf{F}^{ij}$, $j = 1, \cdots, n^c$, where i is the particle in consideration and j the neighbour index ranging from 1 to the number of elements or spheres n^c being in contact with it and (iii) all forces and moments resulting from external damping, $\mathbf{F}^{\text{damp}}$ and $\mathbf{T}^{\text{damp}}$, respectively, which can be written as

$$\mathbf{F} = \mathbf{F}^{\text{ext}} + \sum_{j=1}^{n^c} \mathbf{F}^{ij} + \mathbf{F}^{\text{damp}} \tag{3}$$

$$\mathbf{T} = \mathbf{T}^{\text{ext}} + \sum_{j=1}^{n^c} \mathbf{r}_c^{ij} \times \mathbf{F}^{ij} + \mathbf{T}^{\text{damp}} \tag{4}$$

where $\mathbf{r}_c^{ij}$ is the vector connecting the centre of mass of the i-th particle with the contact point $\boldsymbol{Pc}^{ij}$ with the j-th parti-

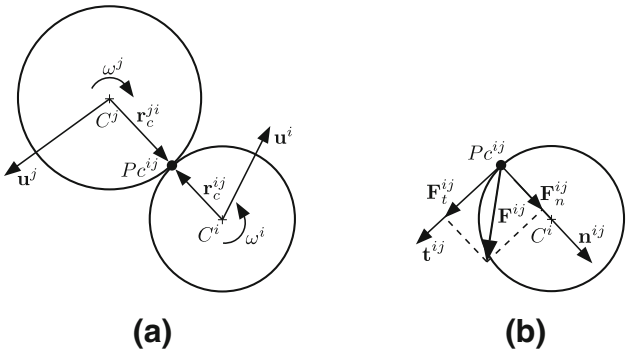


Fig. 1 Decomposition of the contact force into normal and tangential components [30]. **a** Contact between particles. **b** Force decomposition

cle. Figure 1 shows the contact forces between two spherical particles [30].

The contact between the two interacting spheres can be represented by the contact forces $\mathbf{F}^{ij}$ and $\mathbf{F}^{ji}$ (Fig. 1), which satisfy ($\mathbf{F}^{ij} = -\mathbf{F}^{ji}$). Each force $\mathbf{F}^{ij}$ is decomposed into the normal and tangential components, $\mathbf{F}_n^{ij}$ and $\mathbf{F}_t^{ij}$, respectively (Fig. 1)

$$\mathbf{F}^{ij} = \mathbf{F}_n^{ij} + \mathbf{F}_t^{ij} = F_n \mathbf{n}^{ij} + \mathbf{F}_t^{ij}, \tag{5}$$

where $\mathbf{n}^{ij}$ is the unit vector normal to the contact surface at the contact point.

The tangential force $\mathbf{F}_t^{ij}$, along the tangential direction $\mathbf{t}^{ij}$ (Fig. 1), can be written as

$$\mathbf{F}_t^{ij} = F_{t_1} \mathbf{t}_1^{ij} + F_{t_2} \mathbf{t}_2^{ij}, \tag{6}$$

where F_{t_1} and F_{t_2} are the tangential force components along the tangential direction $\mathbf{t}_1$ and $\mathbf{t}_2$, respectively.

The contact forces F_n, F_{t_1} and F_{t_2} are obtained using a constitutive model formulated for the contact between two DE or a DE and a rigid FE. There is a large amount of available contact models [32,33,37] which are compatible with the contact algorithm presented in this article. The examples included will be performed using the formulation summarized in [5]. Standard constitutive models in the DEM are characterized by the normal (k_n) and tangential (k_t) stiffness, normal (d_n) and tangential (d_t) local damping coefficients at the contact interface and Coulomb friction coefficient (μ) represented schematically in Fig. 2 for the case of two discrete spherical particles [32]. Similar models are usually applied for the Sphere-FE contact.

Equations (1) and (2) can be integrated in time using a simple Central Differences scheme [30]. The translational motion at the $(n+1)$-th time step is calculated as follows:

$$\ddot{\mathbf{u}}^n = \frac{\mathbf{F}^n}{m}, \tag{7}$$

$$\dot{\mathbf{u}}^{n+\frac{1}{2}} = \dot{\mathbf{u}}^{n-\frac{1}{2}} + \ddot{\mathbf{u}}^n \Delta t \tag{8}$$

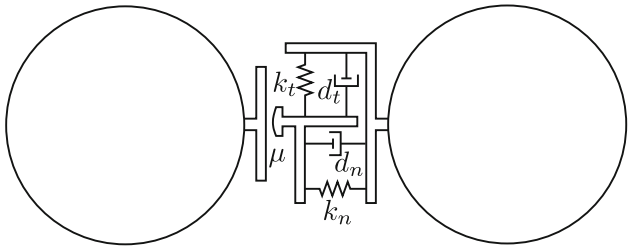


Fig. 2 DEM standard contact interface model [31]

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \dot{\mathbf{u}}^{n+\frac{1}{2}} \Delta t. \tag{9}$$

The integration scheme for the calculation of the increment of rotational motion in each time step ,$\Delta\theta$, is

$$\dot{\boldsymbol{\omega}}^n = \mathbf{I}^{-1} \mathbf{T}^n, \tag{10}$$

$$\boldsymbol{\omega}^{n+\frac{1}{2}} = \boldsymbol{\omega}^{n-\frac{1}{2}} + \dot{\boldsymbol{\omega}}^n \Delta t \tag{11}$$

$$\Delta\boldsymbol{\theta} = \boldsymbol{\omega}^{n+\frac{1}{2}} \Delta t. \tag{12}$$

Explicit integration in time yields high computational efficiency and enables the solution of large models. The disadvantage of the explicit integration scheme is its conditional numerical stability, imposing the limitation on the maximum time step Δt which is determined by the highest natural frequency of the system [46].

2.2 DE/FE contact

The main focus of this work is on the dynamics of the particles (DEs) contacting rigid boundaries (FEs). A single force for every contact entity (facet, edge, vertex) suffices to describe the physics on the particle side but, in general, not on the FE side where deformable solids must be characterized. The contact forces can be transferred to the finite element nodes through interpolation based on FE shape functions [17]. For linear triangles, a simple linear interpolation is used as

$$\mathbf{f}_i = N_i(\mathbf{x}_{P_c})\mathbf{F}, \; N_i(\mathbf{x}_{P_c}) = \frac{\Delta_i}{\Delta_1 + \Delta_2 + \Delta_3}, \; i \in [1,3] \tag{13}$$

where $\mathbf{x}_{P_c}$ is the position of the contact point and Δ_i is the area of the triangle formed by the contact point and the opposite vertices to vertex i in the FE (Fig. 3).

2.2.1 Non-smooth contact

The application of constitutive contact laws such as the Hertz-Mindlin [25] requires that the contact surfaces are smooth and present a unique normal at each point. In the DE/FE contact, usually, the original geometry presents regions where this requirement is not fulfilled. Moreover, even the smooth surfaces loose this feature when they are discretized by means

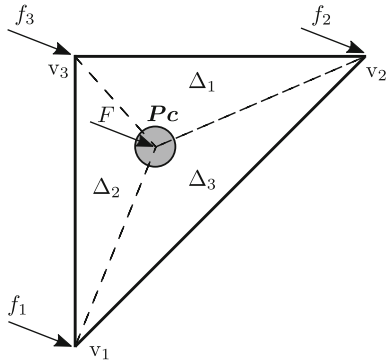


Fig. 3 Punctual Force F and its nodal interpolation to a three-noded triangle by means of linear shape functions expressed using its area coordinates

of FEs. In these situations, a special treatment of the non-smooth regions should be applied under the requirement of some conditions to ensure reasonable results. The following conditions were also analysed in the work by Wellmann [39]:

- The contact constitutive model will be applied normally when the contact is on the facet and will vanish when there is no interpenetration between the elements.
- There should be no discontinuities in the contact force when a contact point evolves from facet to edge and the other way round in order to avoid unphysical results and numerical instabilities.
- The energy should be conserved in an elastic frictionless impact.

The use of the present contact determination algorithm helps the selected contact model ensuring these objectives. Additionally we want this method to yield a result which is geometrically dependent and never mesh dependent. This can be practically achieved as it will be shown in the following Sect. 6 with the limitations detailed in Sect. 7.

2.2.2 Extension to deformable FE

The direct application of hierarchy-based contact algorithms for deformable solids yields not only inaccurate results when the particles come into contact with non-smooth parts of the discretized surface but also instabilities caused by sudden appearance of contact forces at the non-smooth transition between FE. Horner et al. [17], Nakashima et al. [28] and Michael et al. [24], just to name a few, have used the classic hierarchical-based algorithms with linear interpolation of forces in tire—soil interaction applications where the FE deformation is accounted. This can be reasonable in cases where the size of the DEs is relatively small compared to the size of the FEs and the penetration is negligible compared to the DEs radius (small deformation). Even though using

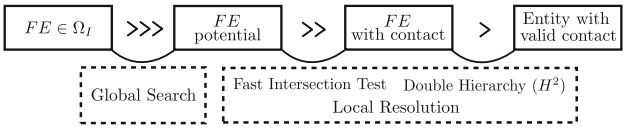


Fig. 4 Neighbour finding scheme. From FE in the intersection domain Ω_I to the valid entities

the H^2 method the major part of these problems is solved, the method is not conceived for this purpose and the fact that it concentrates the contact force in one point is a clear disadvantage.

In general, situations where detailed analysis of strain and stress is conducted, more accurate schemes should be used. Han et al. [14,15] and Wellmann [39] present some algorithms for this purpose.

3 DE/FE contact search algorithm

The DE/FE contact search differs from the DE/DE in the sense that the search is between two different groups of geometries whose intersection is generally more complex and computationally more expensive than the trivial sphere–sphere case. It involves, in general, the resolution of a non-linear system of equations (see the case of superquadrics [4,39] or polyhedra [3,10,29]). The choice of spheres for the DEs and triangles or quadrilaterals for the FEs is a clever choice since this particular check can be done in an efficient way. The contact search is also split into a Global Neighbour Search and a Local Contact Resolution. Our strategy is based on a further split of the Local Contact Resolution in two stages: (a) A Fast Intersection Test and (b) The Double Hierarchy Method. The stages are schematically shown in Fig. 4.

3.1 Global search algorithm

The main purpose of the global search is to determine through a fast rough search which is the potential neighbours for every element in the domain. A basic cell-based algorithm [42] is chosen here which has been parallelized in OMP and adapted for the DE/FE search. The FE domain is selected to build the Search Bins taking advantage of the fact that usually the spatial distribution of the FEs is more regular and in some cases fixed. The algorithm has an additional feature that the Search Bins are built dynamically considering only the FEs belonging to the intersection of the bounding boxes of the DEs and FEs domains. Figure 5a shows how the intersection evolves as long as the simulation goes on. On the other hand, only the DEs inside the intersection domain (Ω_I) will look for their neighbours. This reduces significantly the contact pairs

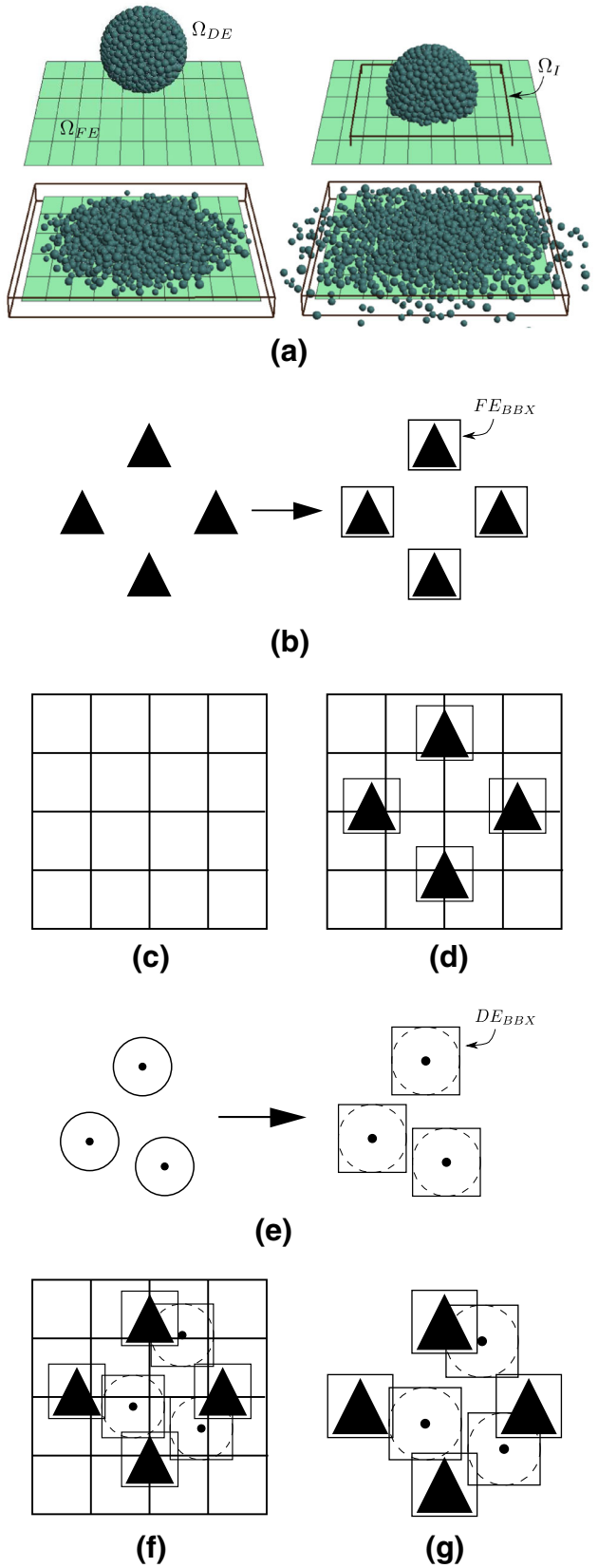


Fig. 5 Sketch showing the search algorithm at the basic contact level. **a** Evolution of the bounding box of the intersection Ω_I. **b** Bounding box of the FEs $\in \Omega_I$. **c** Bins over FEs $\in \Omega_I$. **d** Hash table. **e** Bounding box of the DEs $\in \Omega_I$. **f** Intersection cells. **g** Local Contact Resolution

to be checked and, therefore, the global search performance is increased.

In the global search, every FE and DE has an associated Bounding Box (FE_{BBX}, DE_{BBX}) that is used to tag the position of the elements on the Search Bins and rapidly check for potential neighbours. This is done using a hash table structure as depicted in Fig. 5d which relates each cell to the bounding box FE_{BBX} that fall into it. Rectangular hexahedral bounding boxes encompassing both types of elements are chosen here.

The steps needed to perform the neighbouring search at the global search level are as follows:

(a) Set the bounding box of the intersection of domains Ω_I (Fig. 5a).
(b) Set the bounding box for every FE $\in \Omega_I$ (Fig. 5b).
(c) Generate the Search Bins based on the size and position of the bounding boxes FE_{BBX} of the FEs $\in \Omega_I$ (Fig. 5c).
(d) Place every FE in the Search Bins (looking at the coordinates of their associated bounding box FE_{BBX}) and build the hash table (Fig. 5d).
(e) Set the bounding box for every DEs $\in \Omega_I$ (Fig. 5e).
(f) For every DE particle $\in \Omega_I$ obtain the potential neighbours in the Search Bins. Check the intersection of the DE_{BBX} with the FE_{BBX} of the FEs lying in the surrounding cells (Fig. 5f).

3.2 Local resolution

In the Fast Intersection Test, we determine which potential FE neighbours are in actual contact with each particle. This has to be fast because there are many potential neighbours in the adjacent cells to be checked. Therefore, all detailed contact computations such as determining the type of contact, the contact point, normal direction, etc. are skipped. On the other hand, a good accuracy in the determination of the contacting neighbours is needed. We want to avoid filling the contact pool with FE which do not have contact and therefore have to be eliminated or treated subsequently. This procedure is described in detail in Sect. 4.

In a second stage, the Double Hierarchy Method takes place as a full contact characterization. It determines the type of contact of every neighbour, which contacts are relevant and which ones have to be removed to avoid instabilities or redundant contact evaluations in non-smooth regions and contact transitions. All the detailed contact characteristics are fully determined at this stage for each one of the valid neighbouring entities.

Even though the H^2 Method can be directly applied as a traditional Local Contact Resolution check, the split gives the code higher modularity, i.e. any other contact characteriza-

tion can be applied for the contacting entities. In our in-house code *Kratos* the split yields also higher efficiency (Sect. 6).

4 Fast intersection test

An efficient algorithm designed to determine the intersection of spheres contacting triangles or planar quadrilaterals is described here. We have adapted some of the procedures existing in the computer graphics bibliography [11,20] to the case where the facet contact (inside of the FE) occurs in a substantial higher frequency compared to edge and vertex geometrical contact types. See [18] where the type of contact frequency (facet, edge, vertex) is determined for different number of particles and relative sizes.

The test works for any planar convex polygons of N sides. For every DE $\in \Omega_I$ we loop over the potential FE neighbours provided by Global Neighbour Searching algorithm. Every FE which has contact is stored in an array for every DE sphere.

4.1 Intersection test with the plane containing the FE

The first check is to determine whether the particle intersects the π^m plane formed by the m-th planar finite element $\textcircled{e}^m$. This is represented in Fig. 6.

The outward-pointing normal of the plane can be calculated with the cross product $\boldsymbol{T}$ of any pair of edges taken counter-clockwise. This can be written in the following form, using the permutation tensor ϵ_{ijk} on two edges formed, for example, by the three consecutive vertices $\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3$:

$$T_i = \epsilon_{ijk}(v_j^2 - v_j^1) \cdot (v_k^3 - v_k^2) \tag{14}$$

which has to be normalized to unit length to obtain the normal to the plane $\boldsymbol{n}$

$$\boldsymbol{n} = \frac{\boldsymbol{T}}{\|\boldsymbol{T}\|} \tag{15}$$

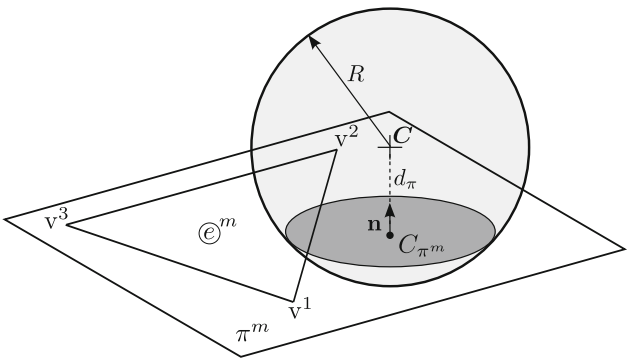


Fig. 6 Intersection of a DE particle with a plane formed by a plane FE

In the case of a zero-thickness wall which can contact for both sides of the FE, the sense of the normal will be set such that points outwards to each particle centre. Once the normal is defined, the distance of the DE centre $\boldsymbol{C}$ to the plane π^m can be determined taking any known point of the plane, namely a vertex v^a, as

$$d_\pi = \sum_{i=1}^{3} \left(n_i \cdot C_i - n_i \cdot v_i^a\right) \tag{16}$$

The distance d_π should be compared to the radius R. If and only if $|d_\pi| \leq R$, the contact between the sphere and the FE is possible. In this case, we proceed with the next checks. Otherwise, the contact with the current FE is discarded and we will jump to check the next potential FE neighbour.

4.2 Inside-Outside test

The purpose of this test is to determine whether the contact is inside the FE (facet contact) or outside (edge, vertex or no contact). It applies to the cases where $|d_\pi| \leq R$. A modification of the Inside-Outside status check [38] is used. The projection C_{π^m} of the centre $\boldsymbol{C}$ of a DE onto the plane π^m formed by an element $\textcircled{e}^m$ with normal n can be calculated as

$$\mathbf{C}_{\pi^m} = \mathbf{C} - d_\pi \cdot \mathbf{n} \tag{17}$$

The next step is to evaluate whether the projection C_{π^m} lies inside or outside the FE $\textcircled{e}^m$ with respect to every edge $\boldsymbol{e}^a$ formed with the vertices $\mathbf{v}^a$ and $\mathbf{v}^{a+1}(\mathbf{v}^N = \mathbf{v}^0)$ (see Fig. 7). For every edge $\boldsymbol{e}^a$, we compute the cross product sign s^a as

$$\boldsymbol{e}^a = \mathbf{v}^{a+1} - \mathbf{v}^a \tag{18}$$

$$s^a = \left(\boldsymbol{e}^a \times (\boldsymbol{C}_{\pi^m} - \mathbf{v}^a)\right) \cdot \mathbf{n} \tag{19}$$

If the product is positive, the projection point C_{π^m} turns to be inside the triangle with respect to that edge. The loop proceeds with the next edges. If the same result is found for every edge, contact occurs with the facet of the FE (inside)

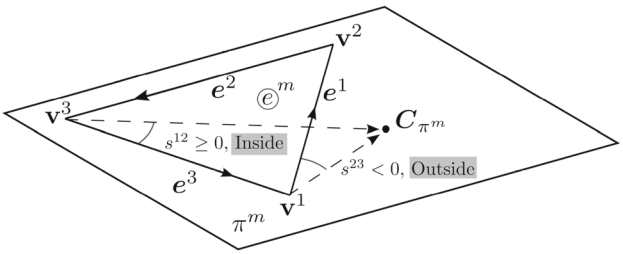


Fig. 7 Inside-Outside check of the projection point edge by edge

and so the contact is assured. Otherwise, if for any edge an *outside* status is found, the loop aborts automatically and no contact with facet can be found. The current value of the edge index a is stored in an auxiliary variable $index_e$ which will be used in the next step where contact with vertices or edges is checked.

4.3 Intersection test with an edge

This test is needed for the cases where $|d_\pi| \leq R$ but the Inside-Outside test failed. Here we use the idea that the edge contact cannot happen to be on the edges where the Inside-Outside check yields a "inside" status. Therefore, it is recommendable to test the edges $\boldsymbol{e}^a$ with $a \in [index_e, N]$ starting from the vertex which failed in the previous test and skipping the previous ones (Note that the edge check is the most expensive one). This approach has also been used by Chen et al. [6].

First, the shortest distance d_e between the edge $\boldsymbol{e}^a$ and the particle centre $\boldsymbol{C}$ should be calculated and compared to the radius R (Fig. 8). The distance is calculated finding out the contact point $\boldsymbol{Pc}$, as

$$d_e = \|\boldsymbol{Pc} - \boldsymbol{C}\| \tag{20}$$

$$\boldsymbol{Pc} = \mathbf{v}^a + p\frac{\boldsymbol{e}^a}{\|\boldsymbol{e}^a\|} \tag{21}$$

$$\boldsymbol{e}^a = \mathbf{v}^{a+1} - \mathbf{v}^a \tag{22}$$

where p is the distance resulting from the projection of the vector connecting the centre $\boldsymbol{C}$ and the vertex $\mathbf{v}^a$ onto the edge $\boldsymbol{e}^a$:

$$p = (\boldsymbol{C} - \mathbf{v}^a) \cdot \boldsymbol{e}^a \tag{23}$$

If $d_e > R$, the contact with this edge is not possible and the check starts again with the next edge $\boldsymbol{e}^{a+1}$. Otherwise, if $d_e \leq R$ we determine where the $\boldsymbol{Pc}$ lies, along the edge, with the help of η, defined as

$$\eta = \frac{p}{\|\boldsymbol{e}^a\|} \tag{24}$$

The case of $0 \leq \eta \leq 1$ implies edge contact. Therefore, contact is found and the Fast Intersection Test finishes yielding a positive result. The FE neighbour is saved to the current DE and the algorithm proceeds to check the next potential FE neighbour.

Otherwise, if this test failed for the current edge $\boldsymbol{e}^a$, the connecting vertices ($\mathbf{v}^a$ and $\mathbf{v}^{a+1}$) have to be evaluated. A value of $\eta < 0$ indicates that the check has to be done with $\mathbf{v}^a$; On the other hand, for $\eta > 1$ the vertex to be tested is $\mathbf{v}^{a+1}$.

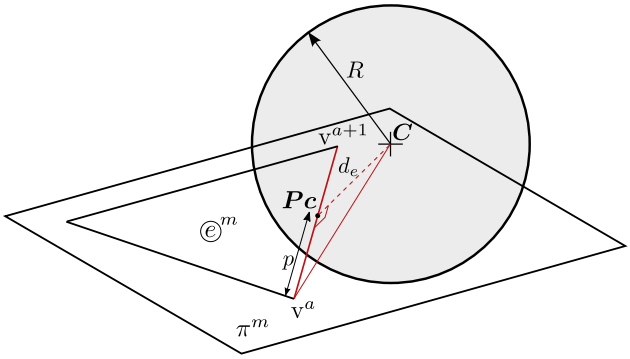


Fig. 8 Intersection of a DE particle with an edge

4.4 Intersection test with a vertex

For the vertex $\mathbf{v}^a$ under consideration, we calculate the squared distance to the DE centre $\boldsymbol{C}$:

$$d_{\mathbf{v}^a}{}^2 = \sum_{i=0}^{i<3} \left(\boldsymbol{C}_i - \mathbf{v}_i^a\right)^2 \tag{25}$$

If $d_{\mathbf{v}^a}{}^2 \leq R^2$, then the Fast Intersection Test yields a positive result and the test finishes. Otherwise the test moves on with the check of the next edge $\boldsymbol{e}^{a+1}$ and its subsequent vertices.

We recall that the purpose of this Fast Intersection Test is merely to determine whether there is intersection or not between the DE sphere and the FE planar convex polygon. An intersection found with a vertex or edge does not assure that this is the actual contact point. In this case, however, we omit at this stage further checks with subsequent edges or vertices where the contact point can happen to be closer.

4.5 Fast Intersection Test scheme

The full algorithm for planar convex polygons of N sides is summarized in Table 2.

5 Double Hierarchy (H^2) Method

This procedure applies only to the list of *FE with contact* that the Fast Intersection Test has generated for every particle. In the case of no previous Fast check this operation could be directly applied as a Local Contact Resolution with the disadvantage that many potential FE have to be tested. It is developed in two different stages:

- *Contact Type Hierarchy (Sect. 5.1)* where for every contacting FE the contact entity with higher priority is determined.
- *Distance Hierarchy (Sect. 5.2)* the elimination procedure takes place determining which contact points have dis-

Table 2 Fast Intersection Test scheme

Parallel loop over all DE, check FE potential neighbours.

(1) Intersection with plane containing the FE $\textcircled{e}^m$

Calculate normal $T_i = \epsilon_{ijk}(\mathrm{v}_j^2 - \mathrm{v}_j^1) \cdot (\mathrm{v}_k^3 - \mathrm{v}_k^2)$.
Normalize it $\boldsymbol{n} = \frac{\boldsymbol{T}}{\|\boldsymbol{T}\|}$ and point it outwards.
Calculate distance to plane $d_\pi = \sum_{i=1}^{3} (n_i \cdot C_i - n_i \cdot \mathrm{v}_i^a)$.
if($|d_\pi| > R$): $\Rightarrow$ **Go to** (4) $(False)$.
else: $\Rightarrow$ Calculate $\mathbf{C}_{\pi^m} = \mathbf{C} - d \cdot \mathbf{n}$ and **Go to** (2).

(2) Inside-Outside test

Initialize $index_e = 0$ and Inside-Outside flag = In.
loop over every edge $\boldsymbol{e}^a = \mathbf{v}^{a+1} - \mathbf{v}^a$ with $a \in [0, N]$.
calculate $s^a = (\boldsymbol{e}^a \times (\boldsymbol{C}_{\pi^m} - \mathbf{v}^a)) \cdot \mathbf{n}$.
if$(s^a < 0)$: $\Rightarrow$ Inside-Outside = Out.
Break loop. Save $index_e = a$. **Go to** (3).
else$(s^a \geq 0)$: $\Rightarrow$ **Continue** with next edge.
if(Inside-Outside flag == In): $\Rightarrow$ **Go to** (4) $(True)$.
else: $\Rightarrow$ **Go to** (3).

(3) Intersection with Edge and Vertex

loop over every edge $\boldsymbol{e}^a$ with $a \in [index_e, N]$.
Calculate projection: $p = (\boldsymbol{C} - \mathbf{v}^a) \cdot \boldsymbol{e}^a$.
Calculate the contact point: $\boldsymbol{Pc} = \mathbf{v}^a + p\frac{\boldsymbol{e}^a}{\|\boldsymbol{e}^a\|}$.
Calculate distance to edge $d_{\boldsymbol{e}} = \|\boldsymbol{Pc} - \boldsymbol{C}\|$.
if$(d_e > R)$: $\Rightarrow$ **Continue** with next edge.
else: Calculate $\eta = \frac{p}{\|\boldsymbol{e}^a\|}$.
if$(0 \leq \eta \leq 1)$: $\Rightarrow$ **Go to** (4) $(True)$.
if$(\eta < 0)$: $\Rightarrow$ $d^2_{\mathrm{v}^a} = \sum_{i=0}^{i<3} (\boldsymbol{C_i} - \mathbf{v}_i^a)^2$.
if$(d^2_{\mathrm{v}^a} \leq R^2)$: $\Rightarrow$ **Go to** (4) $(True)$.
else: $\Rightarrow$ check next edge.
if$(\eta > 1)$: $\Rightarrow$ $d^2_{\mathrm{v}^{a+1}} = \sum_{i=0}^{i<3} \left(\boldsymbol{C_i} - \mathbf{v}_i^{a+1}\right)^2$.
if$(d^2_{\mathrm{v}^{a+1}} \leq R^2)$: $\Rightarrow$ **Go to** (4) $(True)$.
else: $\Rightarrow$ check next edge.
Go to (4) $(False)$.

(4) Contact Found $(True/False)$

$True$: $\Rightarrow$ Store $\textcircled{e}^m$ as *FE with contact* and **Continue**.
$False$: $\Rightarrow$ **Stop!** No contact.

tance priority over others which are redundant or false and have to be eliminated.

5.1 Contact Type Hierarchy

The basis of this procedure is that each primitive has hierarchy over its sub-entities, i.e. a facet of a N-sides polygon has hierarchy over the N edges that compose it. In turn each of the edges $\boldsymbol{e}^a$ has hierarchy over its two vertices $\mathbf{v}^a$, $\mathbf{v}^{a+1}$. Figure 9 outlines the Contact Type Hierarchy for a triangle.

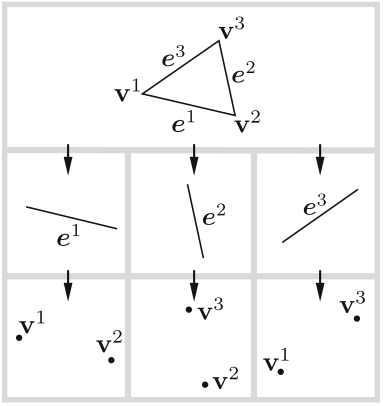


Fig. 9 Contact Type Hierarchy for a triangle. The facet has higher hierarchy than its edge and vertices. The edges have higher hierarchy than its vertices

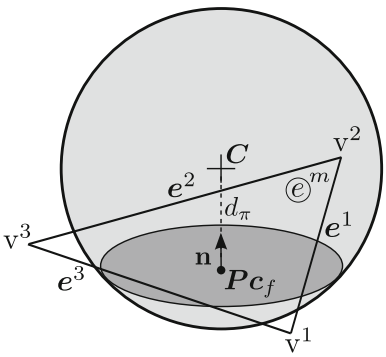


Fig. 10 Contact with facet. Edges and vertices are discarded from contact check

The algorithm is organized as a sequence of three entity-checking levels. If a particle is in contact with the facet of a FE the contact search over its edges and vertices, which are in a lower hierarchy level, is discarded (see Fig. 10). Otherwise, if contact with the FE facet does not exist, the contact check should continue over the sub-entities. Similarly, at the edges level, any contact with an edge cancels out further contact checks for those two vertices belonging to that edge. It does not cancel out, however, the contact check with the other edges because they are at the same hierarchy level. Table 3 in Sect. 5.1.4 displays the pseudocode of the contact Type detection.

Every time a new contact entity is determined by the Contact Type Hierarchy, the Distance Hierarchy (Sect. 5.2) takes place immediately after. The Distance Hierarchy will determine if the new contacting entity found is redundant or non-valid, if it cancels out the previously found ones or if it is a new valid contacting entity to be considered for the DE.

Table 3 Contact Type Hierarchy algorithm

Loop over every FE neighbour with contact $\textcircled{e}^m$.

(1) Facet level
- Project the centre onto the plane $\mathbf{C}_{\pi^m}$ (Eq. 17).
- Perform the Inside-Outside test (Sect. 4.2)
- **if** Contact: $\Rightarrow$
  - **Go to** Distance Hierarchy (Table 4) and **Stop!**
- **else**: $\Rightarrow$ **Go to** (2) with index $index_e$.

(2) Edge level
- **loop** over every edge $\boldsymbol{e}^a$ with $a \in [index_e, N]$.
  - Perform the Edge Check (Sect. 4.3).
  - **if** Contact $\Rightarrow$ **Go to** Distance Hierarchy (Table 4).
  - **else if** ($d_e \leq R$ and $\eta < 0$) $\Rightarrow$ **Go to** (3) with $\mathbf{v}^a$.
  - **else if** ($d_e \leq R$ and $\eta > 1$) $\Rightarrow$ **Go to** (3) with $\mathbf{v}^{a+1}$.
  - **Continue** with the next edge.

(3) Vertex level
- Perform Vertex check (Sect. 4.4).
- **if** Contact $\Rightarrow$ **Go to** Distance Hierarchy (Table 4).
- **Go To** Edge level and check next edge.

For any valid contact entity the geometrical contact characteristics that will be stored are as follows:

- The contact Point $\boldsymbol{Pc}$.
- The FE nodal weights.
- The contact type: Facet, Edge or Vertex.

Note that some of the geometrical characteristics such as the distance, the normal vector or the contact local axis can be recalculated later when the contact constitutive law is applied and, thus, it is optional to store them here at this stage.

5.1.1 Facet level

The check proceeds in the same way as explained in Sect. 4, checking for the intersection of the DE with the plane formed by the FE (Sect. 4.1). If the Fast Intersection Test has been performed previously $|d_\pi| \leq R$ is necessarily true since contact has been found for this FE. Otherwise, if no previous Fast Intersection Test has been carried out, this condition applies to discard FE without contact.

Next, the Inside-Outside test (Sect. 4.2) has to be performed. This test will tell us whether the projection C_{π^m} (Eq. 17) lies on the facet (inside the FE) or it is outside, contacting with the edges or vertices. Figure 11 shows two examples where the projection C_{π^m} is inside and outside the FE facet.

The values of the cross product sign s^a obtained from Eq. 19 for every edge $\boldsymbol{e}^a$ are used to obtain the weights of the shape function at the contact point. The areas needed for the calculation are simply one half of the cross product sign: $\Delta_a = s^a/2$. The weights of the nodal shape functions on the contact point are then calculated. For a triangle:

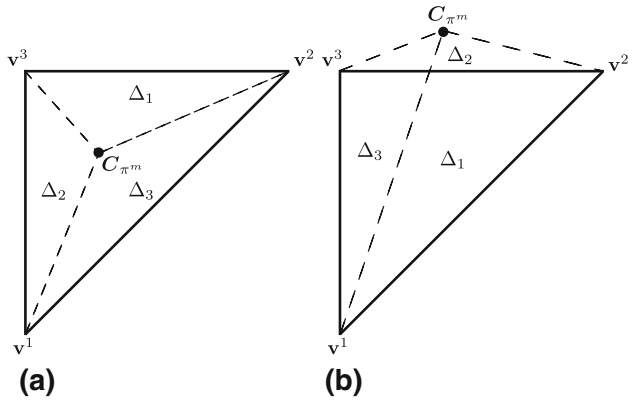


Fig. 11 Example of projection C_{π^m} inside the FE facet (**a**) (facet contact) and outside (**b**) (no facet contact) for a triangle. **a** C_{π^m} inside the facet. **b** C_{π^m} outside the facet.

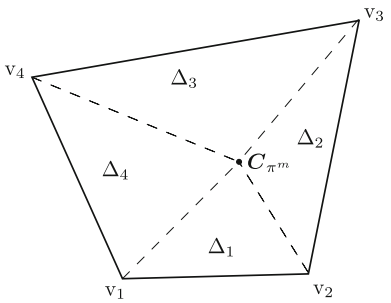


Fig. 12 Triangular areas for the calculation of shape function values in a planar convex quadrilateral

$$N_1 = \frac{\Delta_2}{\hat{\Delta}_T}, \quad N_2 = \frac{\Delta_3}{\hat{\Delta}_T}, \quad N_3 = \frac{\Delta_1}{\hat{\Delta}_T} \tag{26}$$

where

$$\hat{\Delta}_T = \Delta_1 + \Delta_2 + \Delta_3 \tag{27}$$

For four-node convex quadrilaterals (Fig. 12 the following expression can be applied as introduced in Zhong [45]):

$$N_1 = \frac{\Delta_2 \Delta_3}{\hat{\Delta}_Q}, \quad N_2 = \frac{\Delta_3 \Delta_4}{\hat{\Delta}_Q}$$
$$N_3 = \frac{\Delta_4 \Delta_1}{\hat{\Delta}_Q}, \quad N_4 = \frac{\Delta_1 \Delta_2}{\hat{\Delta}_Q} \tag{28}$$

where

$$\hat{\Delta}_Q = (\Delta_1 + \Delta_3)(\Delta_2 + \Delta_4) \tag{29}$$

Note that if any of the cross product signs s^a evaluated with respect to the edge $\boldsymbol{e}^a$ yields a negative value the check

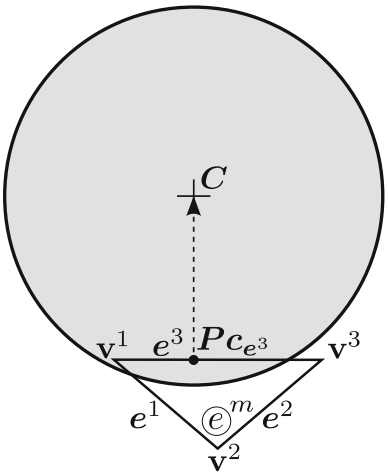


Fig. 13 Contact with edge. Vertices belonging to that edge are discarded from contact check

stops since the projection of the centre C_{π^m} lies outside. The current edge index $index_e$ is stored and it will be the first to be checked as it has been appointed in Sect. 4.3.

If the projection C_{π^m} (Eq. 17) lies inside the facet, it becomes the contact point $\boldsymbol{Pc}$. Due to the highest hierarchy level of the facet, the Contact Type Hierarchy finishes here for this FE. The Distance Hierarchy is now called and all the necessary contact characteristics are saved.

5.1.2 Edge level

At this level the edge check detailed in Sect. 4.3 has to be applied for every edge $\boldsymbol{e}^a$ with $a \in [index_e, N]$ starting with the first edge that yielded an outside status at the Facet level.

When contact with the edge $\boldsymbol{e}^a$ is found the check at the lower level for the vertices associated to it, $\mathbf{v}^a$ and $\mathbf{v}^{a+1}$, is discarded (Fig. 13). The contact check with the following edges cannot be discarded, however, since they are at the same hierarchy level in terms of Contact Type. The Distance Hierarchy will determine the validity of the new contact and eliminate or substitute previous ones. This is a key difference with the Fast Intersection Test where the check automatically stops once a contact entity is found.

The nodal weights can be obtained from the η parameter (Eq. 24) at the edge $\boldsymbol{e}^a$. Figure 14 shows graphically how η is determined,

$$N_a = 1 - \eta, \quad N_{a+1} = \eta \quad (N_N = N_0) \tag{30}$$

Equation 30 shows the values at the nodes connected to the edge $\boldsymbol{e}^a$. The rest of nodes have a null value for its shape functions.

If the edge contact check failed but the distance d_e (Eq. 20) is lower than the radius ($d_e \leq R$) the closest vertex (based on the calculation of η) will be checked. The check will proceed

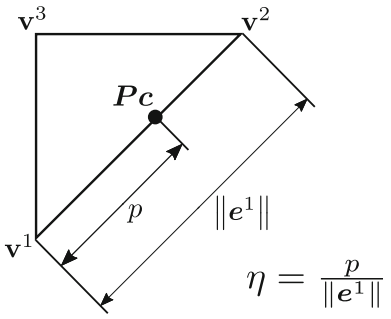


Fig. 14 Weights for an edge contact in a triangle

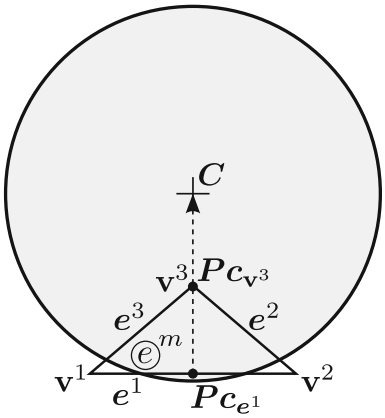


Fig. 15 Contact with edge and vertex. When contact exist with edge $\boldsymbol{e}^3$ it can also exists with vertex $\mathbf{v}^2$

in any case (found edge, found vertex or none) with the next edges.

5.1.3 Vertex level

The vertex check is described in Sect. 4.4. Figure 15 illustrates why the edge $\boldsymbol{e}^a$ has hierarchy over its two vertices $\mathbf{v}^a$, $\mathbf{v}^{a+1}$ but not over the non-contiguous one $\mathbf{v}^{a+2}$. The shape function weights are 1 for the found vertex and 0 for the rest.

As usual the Distance Hierarchy is called after the contact is detected and, if the contact is valid, its characteristics are stored.

5.1.4 Contact Type Hierarchy scheme

The scheme of Table 3 assumes that the Fast Intersection Test has taken place already. For every DE the first loop is over the found neighbours. The check can be performed in parallel for every particle in the model.

5.2 Distance Hierarchy

A spherical particle can be, in general, in contact with many different FE entities. Sometimes these contacts are result of the penetrations introduced by the penalty method and some

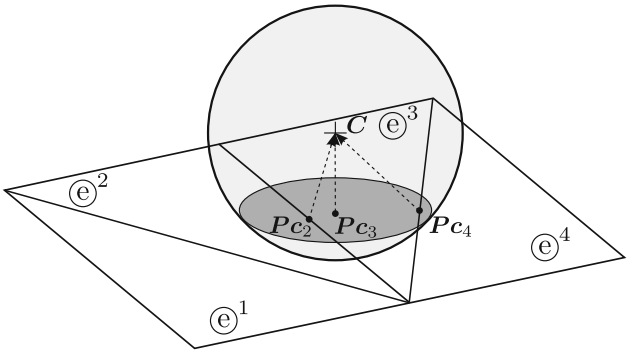


Fig. 16 Contact between a DE and a FE mesh whose elements are smaller than the indentation

contacts give redundant or invalid information and, therefore, should be eliminated. This is the scenario shown in Fig. 16 where contact with elements ⓔ^{2}, ⓔ^{3} and ⓔ^{4} is detected. In a collision of the sphere normal to the plane, the force applied by the plane surface to the sphere must have also a normal direction and a magnitude only given by the penetrations and independent of the position x and y on the plane. Therefore, the contact force coming from the edges of elements ⓔ^{2} and ⓔ^{4} should not be taken into account. This is solved by the distance-based hierarchy which is an elimination procedure that takes place every time a new contact entity is found at the Contact Type Hierarchy.

The procedure basically compares the contact vectors against their projections one another. The new contact vector $\mathbf{Vc}_i = \mathbf{C} - \boldsymbol{Pc}_\mathbf{i}$ is projected onto the previously found contact vector $\mathbf{Vc}_j = \mathbf{C} - \boldsymbol{Pc}_j$ and vice versa. The following expressions are obtained:

$$Pr_{i,j} = \mathbf{Vc}_i \cdot \frac{\mathbf{Vc}_j}{\|\mathbf{Vc}_j\|}, \quad Pr_{j,i} = \mathbf{Vc}_j \cdot \frac{\mathbf{Vc}_i}{\|\mathbf{Vc}_i\|} \tag{31}$$

The contact check is performed using the algorithm presented in Table 4:

Table 5 shows an example of how the elimination procedure is performed for two different possible cases. On the left side all the found contact vectors are represented. A graphical interpretation of the projections is also given for the first example. On the right side, only the final relevant contact vectors, that the Distance Hierarchy yields, are shown.

In the first situation, no contact with edges of elements ⓔ^{2} and ⓔ^{4} is taken into account, since their projections, $Pr_{2,3}$ and $Pr_{4,3}$, over the facet contact vector of element ⓔ^{3} have the same module as the contact vector $\mathbf{Vc}_3$ itself.

In the second situation, the sphere has contact with the facet of element ⓔ^{4}, the edge of element ⓔ^{3} and the shared edge of elements ⓔ^{1} and ⓔ^{2} which will be appearing as two different contact vectors $\mathbf{Vc}_1$ and $\mathbf{Vc}_2$ given by the Contact Type Hierarchy stage. These vectors do not appear directly in the figures in Table 5 but they are calculated by $\mathbf{C} - \mathbf{Pc_1}$ and $\mathbf{C} - \mathbf{Pc}_2$, respectively. First, note that either contact with $\mathbf{Vc}_1$ or $\mathbf{Vc}_2$ will be arbitrarily discarded by the elimination procedure since they are mathematically the same vector. Let us assume the $\mathbf{Vc}_1$ is kept and $\mathbf{Vc}_2$ discarded. On the other hand, the projection $Pr_{3,4}$ of the contact vector $\mathbf{Vc}_3$ over the contact vector $\mathbf{Vc}_4$ discards contact with element ⓔ^{3}. Finally, contacts with element ⓔ^{4} and ⓔ^{1} do not discard each other since their projections one another have a value of $Pr_{1,4} = 0$ and $Pr_{4,1} = 0$ (they form a 90° angle) and therefore are smaller than the contact vectors length. Hence both contacts are taken into account, as it is expected.

Table 4 Distance Hierarchy check

Given a new found contact i by the Contact Type Hierarchy:
(1) **loop** over every existing contact ($j = 1, \ldots, n$)
Project $\mathbf{Vc}_i$ on $\mathbf{Vc}_j$:⇒ $Pr_{i,j} = \mathbf{Vc}_i \cdot \frac{\mathbf{Vc}_j}{\|\mathbf{Vc}_j\|}$
Project $\mathbf{Vc}_j$ on $\mathbf{Vc}_i$:⇒ $Pr_{j,i} = \mathbf{Vc}_j \cdot \frac{\mathbf{Vc}_i}{\|\mathbf{Vc}_i\|}$
if ($Pr_{i,j} \geq \|\mathbf{Vc}_j\|$) :⇒ i is an invalid contact.
Go to (2) ($False$) and **break** loop.
else if ($Pr_{j,i} \geq \|\mathbf{Vc}_i\|$) :⇒ j is an invalid contact.
Discard j ! **Continue** loop.
Go to (2) ($True$).
(2) Valid contact ($True/False$)
if ($True$) :⇒ i is valid contact! Save contact details.
else ($False$) :⇒ i is an invalid contact! Discard i!.

The main advantage of this method lies in its wide generality. It works fine for most of the traditional conflictive situations where multi-contacts and FE transitions are present. It is consistent and so the order in which the neighbours have been found and stored does not affect the final result. The tests carried out in the validation (Sect. 6) show that the force vector always has the appropriate direction.

5.3 Note on types of FE geometries

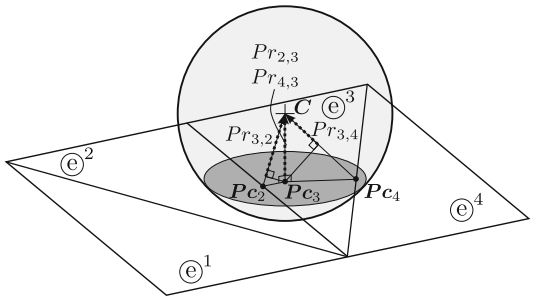
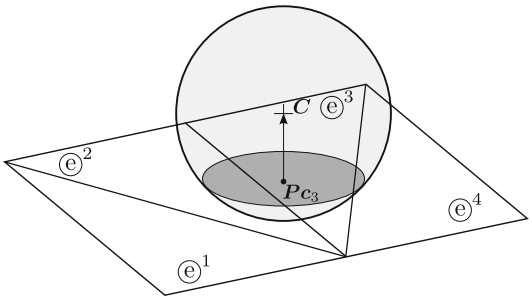
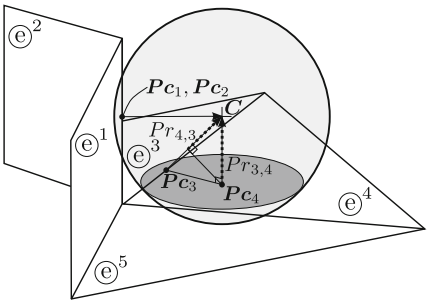
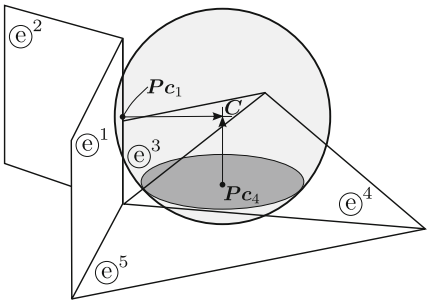
Taking advantage of the generality of the method, the full algorithm can be applied directly to any N-sided planar convex polygonal FE. The weights can be calculated with the barycentric coordinates as introduced in Meyer et al. [23] and further analysed by Sukumar [36]:

$$N_i = \frac{\cot(\alpha_i) + \cot(\beta_i)}{\|\boldsymbol{P}_c - \mathbf{v}_i\|^2} \tag{32}$$

The definition of α_i and β_i is shown in Fig. 17.

Contact surfaces with non-planar quadrilaterals or other curved elements are not on the scope of this paper. Generally it involves a minimization problem [43]. However, Chen [6] proposes an averaging of the normal and a relaxed contact criterion.

Table 5 Contact elimination cases

	Found Contact Points and Vectors	Relevant Contact Vectors
Situation 1		
Situation 2		

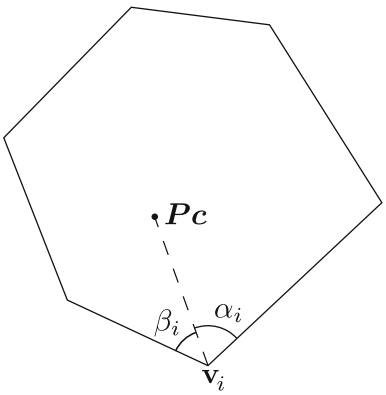


Fig. 17 Angles formed with the vector $v_i - P_c$ and each of the two edges connecting on node i in a polygon

5.4 Note on types of DE geometries

Industrial applications often make use of more accurate strategies to model the particles rather than using classical spherical particles. Among the more popular ones are the superquadrics [39], level set functions [1], or cluster of spheres [13]. Our choice is to model the particles with the sphere clustering technique which provides a solution with a good ratio between accuracy and computational cost and it adapts perfectly to the presented algorithm and, thus, it makes the contact search fast and parallelizable applying the explained technique.

6 Validation

In this section, several examples are carried out to test different aspects of the algorithm performance. First, several examples are carried out to test its performance in critical situations where most of the literature methods would fail. Next, an industrial application example serves to test the method in terms of scalability and parallel computation.

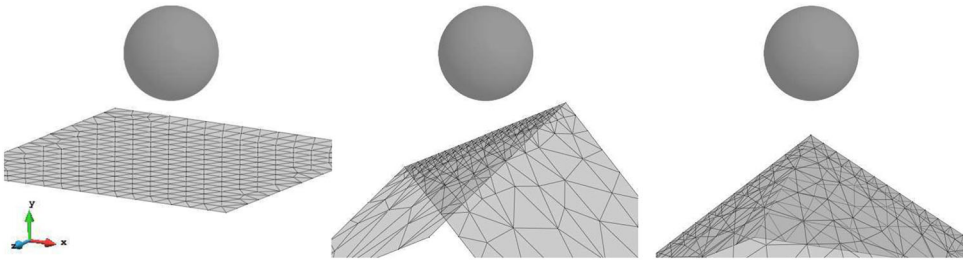
6.1 Benchmarks of critical situations

The following tests do not correspond to practical situations, but they serve instead to validate the contact calculation procedure. All benchmarks have been carried out using a Hertzian contact law [5].

6.1.1 Facet, edge and vertex contact

These first three benchmarks are represented by a sphere, which has low stiffness in order to achieve large indentation, contacting three different boundaries meshed with triangles. In every case, the sphere falls from the same height (1 m)

Fig. 18 Benchmarks layout



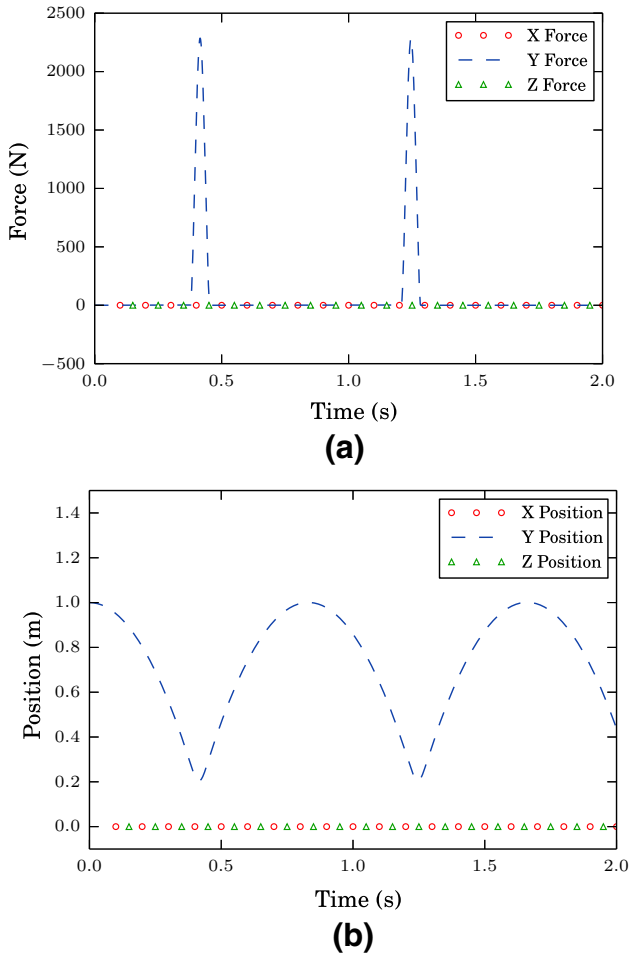


Fig. 19 Results obtained in the benchmark tests. **a** Force exerted by the FEs to the DE. **b** Position of the centre of the DE

vertically and perpendicular to the contact entity which is, respectively, a facet, an edge or a vertex. Figure 18 shows the benchmarks display and Table 6 the simulations parameters.

Since there is no damping applied, the energy should be conserved and the ball must return to the initial position after the rebound. The sphere is expected to follow a vertical trajectory with identical results for the three cases.

Graph in Fig. 19a shows that, although the indentation is greater than the 30 % of the DE radius leading to multiple contacts with all kind of entities, the force is applied only in the vertical direction (Y direction). From this, it can be concluded that the contact elimination procedure performs correctly. The results are exactly the same in the three different scenarios (facet, edge and vertex contact). It verifies also that there is no energy gain or dissipation since the rebound maximum height is the same always as it can be observed in Fig. 19b. This is a good test to see that the method works properly for normal contacts of all three types: with facet, with edge and with vertex independently of the mesh and the indentation achieved (always lower than the radius).

6.1.2 Continuity of contact

It is essential to ensure continuity of the contact force in the non-smooth contact regions and FE element transitions. In the following example, the continuity of the normal force is presented. A DE is set to move along the boundary and its contact transfers from the surface of a triangular element (facet contact) to one of its edges or vertices. A frictionless and rotation-free sphere has a trajectory path enforced (as shown in Fig. 20) so that the indentation is always constant (0.01 m either in contact with the facets $\boldsymbol{f}^1$ and $\boldsymbol{f}^2$ or with the edge $\boldsymbol{e}$). The simulation parameters are the ones presented in Table 6.

If continuity is met, the force module must always be the same. The direction of the contact force should evolve from vertical (normal to $\boldsymbol{f}^1$) to horizontal (normal to $\boldsymbol{f}^2$) with a smooth transition. This is achieved due to the fact that the algorithm gives higher hierarchy to the edge and the vector is calculated joining the contact point and the centre of the sphere (Fig. 21).

The results show that no discontinuities arise when the contact evolves from facet contact to edge contact and vice versa, being the contact force constant along all the simulation and equal to 76.063 N, as expected.

The continuity of the normal forces in a concave transition and the tangential forces across different elements is not fully assured. Even though the error is very small for practical situations, it is something which we consider important to quantify and be aware of. This is reported in Sect. 7.

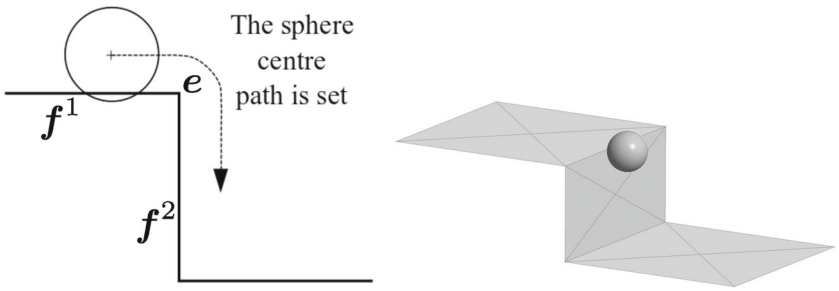


Fig. 20 Simulation scheme

Table 6 Simulation parameters

Material properties	
Radius (m)	0.3
Density (kg/m^3)	100
Friction coefficient DE/FE	0.3
Young modulus (Pa)	10^5
Poisson ratio	0.2
Calculation parameters	
Initial velocity (DE) (m/s)	[0.0, 0.0, 0.0]
Gravity (m/s^2)	[0.0, −9.81, 0.0]
Time step (s)	10^{-5}
Neighbour search frequency	5

6.1.3 Multiple contact

The goal of this test is to check that the method determines correctly the case of a sphere contacting more than one element. The set up of the example consists of three spheres falling onto a plane with three different shape holes, as shown in Fig. 22a. Simulation parameters are presented in Table 7. In this example, damping is applied.

Graph in Fig. 23 shows the velocity modulus of each of the DEs involved in the simulation. It can be seen that the spheres velocity after 2.5 s of simulation is close to 0, as expected and a final equilibrium position is reached for every sphere involving simultaneous contacts with vertices and edges.

6.1.4 Mesh independence

The following example simulates a ball sliding on a plane with friction. The sphere is set in vertical equilibrium upon the plane and a horizontal velocity is imposed. The sphere should start sliding, while its angular velocity will progressively increase up to a constant value at which the sliding event finishes and only rolling occurs thereafter. This is schematically depicted in Fig. 24a.

The analytical solution can be calculated to validate the simulation using equilibrium equations with kinematic compatibility conditions and the basic Coulomb friction law. The moment of inertia of a sphere is $I_\theta = 2/5mR^2$. The following is obtained for the combined sliding and rotation phase:

$$v(t) = v_0 - \mu g t \tag{33}$$

$$x(t) = v_0 t - 1/2\mu g t^2 \tag{34}$$

$$\omega(t) = \frac{R\mu mg}{I_\theta} t = \frac{5\mu g}{2R} t \tag{35}$$

Equation 35 comes from integrating the angular acceleration $\dot{\omega}$ for the case zero initial angular velocity. The constant rolling event occurs when the tangential velocity v matches the angular velocity ω times the radius R:

$$\begin{aligned} v &= R\omega \\ t_c &= \frac{2v_0}{7\mu g} \end{aligned} \tag{36}$$

For time $t > t_c$, the equations of motion are as follows:

$$v(t) = \frac{5}{7} v_0 \tag{37}$$

$$x(t) = \frac{12 v_0^2}{49\mu g} + \frac{5}{7} v_0 (t - t_0) \tag{38}$$

$$\omega(t) = \frac{5v_0}{7R} \tag{39}$$

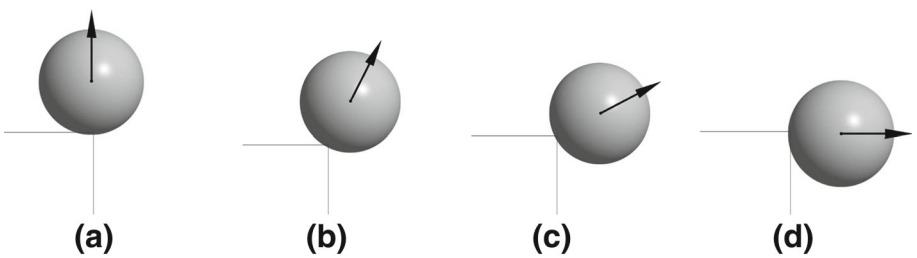


Fig. 21 Force applied by the surface and the edge to the sphere at different instants of the simulation. **a** Contact f^1. **b** Contact e. **c** Contact e. **d** Contact f^2

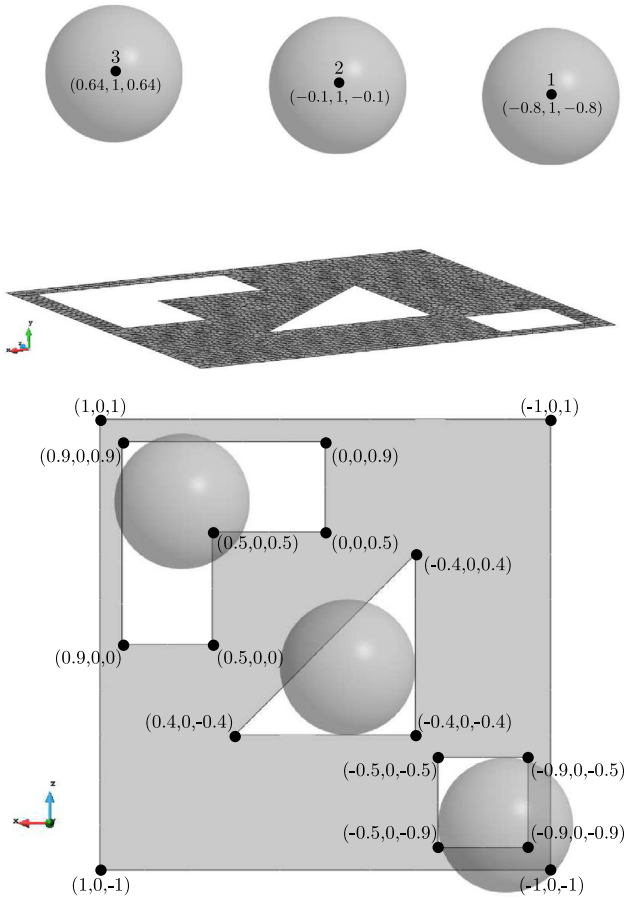


Fig. 22 Multiple contact test geometry

Table 7 Simulation parameters

Material properties	
Radius (m)	0.3
Density (kg/m^3)	100
Friction coefficient DE/FE	0.3
Young modulus (Pa)	10^6
Poisson ratio	0.2
Restitution coefficient	0.4
Calculation parameters	
Initial velocity (DE) (m/s)	[0.0, 0.0, 0.0]
Gravity (m/s^2)	[0.0, −9.81, 0.0]
Time step (s)	10^{-5}
Neighbour search frequency	1

The set up of the simulation is shown in Fig. 24b. Two cases are compared, one involves sliding on a plane discretized by a single quadrilateral element, while in the other case the plane is discretized by 80 triangular elements. The parameters of the simulation are the same as in the previous example, detailed in Table 7. The spheres are given an initial velocity of $5m/s$ in the x direction. The simulation has been run for 1 s. The simulation results are plotted together with the analytical solution in Fig. 25.

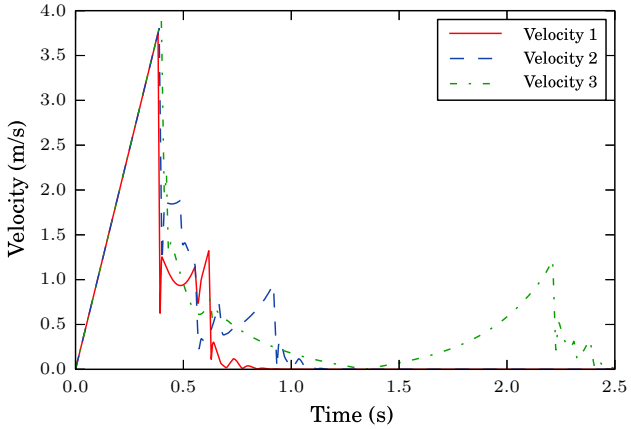


Fig. 23 DEs velocity

Only one numerical solution was included in the plot of Fig. 25 since the difference between meshes turned to be negligible. In Table 8 the values of the displacement (x), velocity (v) and angular velocity (ω) at the end of the simulation ($t = 1$) are presented.

This example shows how the results on the DE practically independent on the boundary mesh selected. On the other side, for the simulation performed, the numerical results agreed perfectly with the theoretical solution. This case does not show any noticeable discontinuity in the normal and tangential contact forces in the transition between boundary FEs. In Sect. 7 a note on this issue is given too.

6.2 Algorithm behaviour on industrial examples

In order to evaluate the overall method behaviour, the simulation of a particle mixer has been carried out. The model represents a rotary mixer where contact occurs between DEs and the three different FE entities (facets, edges and vertices) of the boundary mesh composed by triangular and quadrilateral elements. Additionally, the simulation has been used to evaluate the parallelization behaviour.

6.2.1 Description of the simulation

Figures 26 and 27 show the geometry and the initial arrangement of the simulation composed by 29559 DEs, 848 triangular FEs and 1600 quadrilateral FEs.

Table 9 presents the material properties and simulation parameters used.

The calculations have been carried out assuming a Hertzian contact law [5]. For the integration of the equations of motion the classical central difference integration scheme has been used [30]. Additionally, in this test, some rolling resistance moment has been added to model the particle irreg-

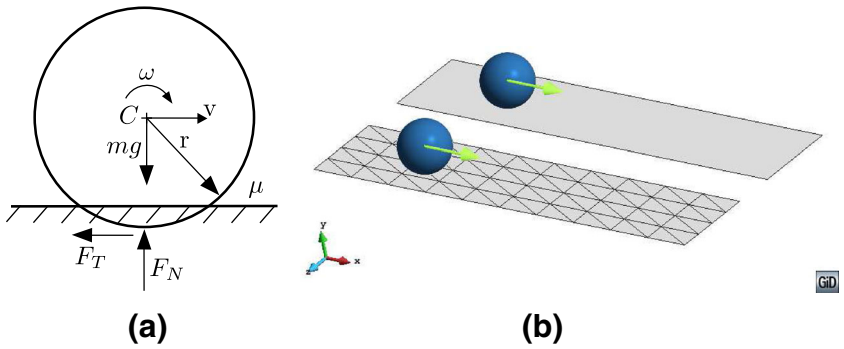


Fig. 24 Benchmark of a sliding sphere on a plane with friction. **a** Problem definition. **b** Simulation set up

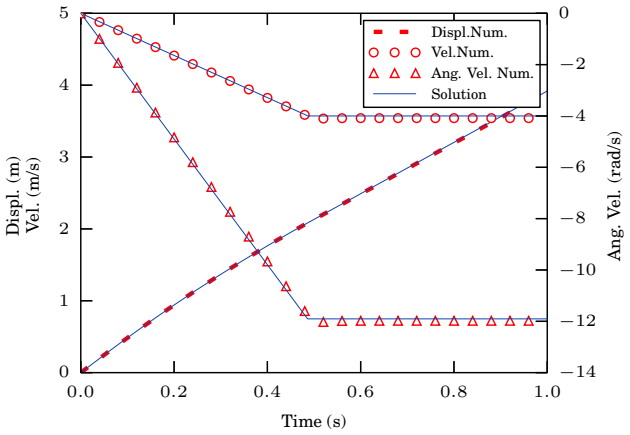


Fig. 25 Numerical results of the displacement and velocity in X with the angular velocity in Z compared against the theoretical solution

Table 8 Results at the end of the simulation

	Quadrilateral	Triangle	Analytical
x (m)	3.9021	3.9022	3.9182
Error (%)	0.4102	0.4071	–
v (m/s)	3.5410	3.5410	3.5714
Error (%)	0.8528	0.8528	–
ω (rad/s)	−11.9788	−11.9788	−11.9048
Error (%)	0.0062	0.0062	–

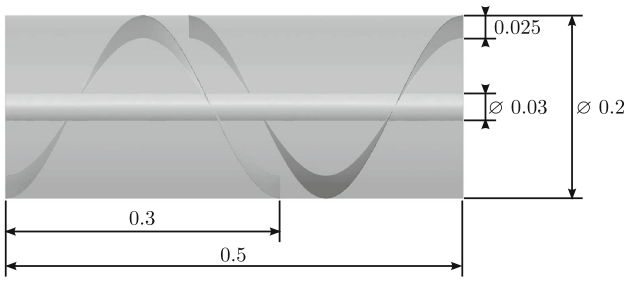


Fig. 26 Geometry of the horizontal rotary mixer. Distance expressed in meters

ularities. In this case, an improvement to the classical rolling resistance model A presented by Wensrich and Katterfeld [40] has been developed, in order to avoid the instabilities that

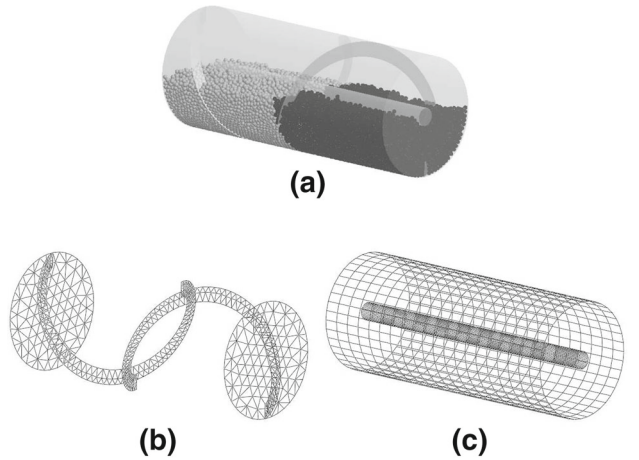


Fig. 27 Mesh used in the horizontal rotary mixer simulation. **a** DEs initial arrangement. **b** Triangular FEs. **c** Quadrilateral FEs

Table 9 Simulation parameters

Material properties	
Radius (m)	0.0035
Density (kg/m^3)	1000
Friction coefficient DE/DE	0.5
Friction coefficient DE/FE	0.75
Young modulus (Pa)	10^7
Poisson ratio	0.2
Rolling friction coefficient	0.001
Restitution coefficient	0.4
Calculation parameters	
Rotation velocity (rad/s)	18.85
Time step (s)	5×10^{-5}
Neighbour search frequency	1
Simulation time (s)	20.0

appear when $\boldsymbol{\omega}^{rel}$ is almost 0. This complete new approach can be found in [19].

6.2.2 *Simulation results and code performance in serial*

The DEMPack code was tested in an Intel Xeon E5-2670. It took 29 h, 20 min and 30 s in serial to run 20 s of simulation

Table 10 Serial performance of the code for the industrial example

	Split fast + H^2	Direct H^2 method
DE/DE contact search (%)	53.9	51.4
DE/FE contact search (%)	20.7	23.9
Create bins and others (%)	4.5	4.2
Fast intersection (%)	15.1	–
H^2 method (%)	1.2	19.7
Total time (s)	105,630	111,041

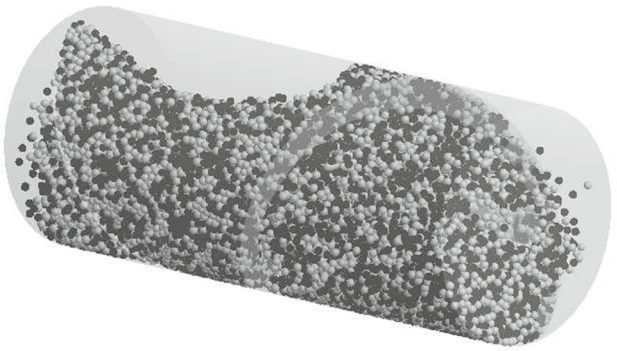


Fig. 28 Spheres arrangement after 20 s

which comprehends 400,000 time steps. Some results about the performance of the code are presented in Table 10. In this specific case, which involves approximately 30k DE and 2.5k FE, it can be seen that the calculation effort for DE/FE contact search represents about the 20 % of the total CPU time. The results showed that by splitting the Fast Intersection and the H^2 Method the code turned to be 5 % faster which is a significant improvement for this case, where most of the contacts are DE/DE. It can be also seen that the cost of the H^2 Method is very low (only 1 %) when the split is applied.

Figure 28 displays the arrangement of the spheres after 20 s. This mixer simulation has been considered due to the great amount of contacts (DE/DE and DE/FE) that it involves.

6.2.3 Code performance in parallel

Graphs in Fig. 29 show the code performance using an OpenMP parallel computing strategy. Based on the results it can be concluded that, despite being the speedup far from the ideal linear case, the fact that the contact check algorithm is totally parallel helps to the performance.

6.2.4 Performance of the search algorithm

An analysis has been performed within this industrial example in order to detect how many FE are treated in each of

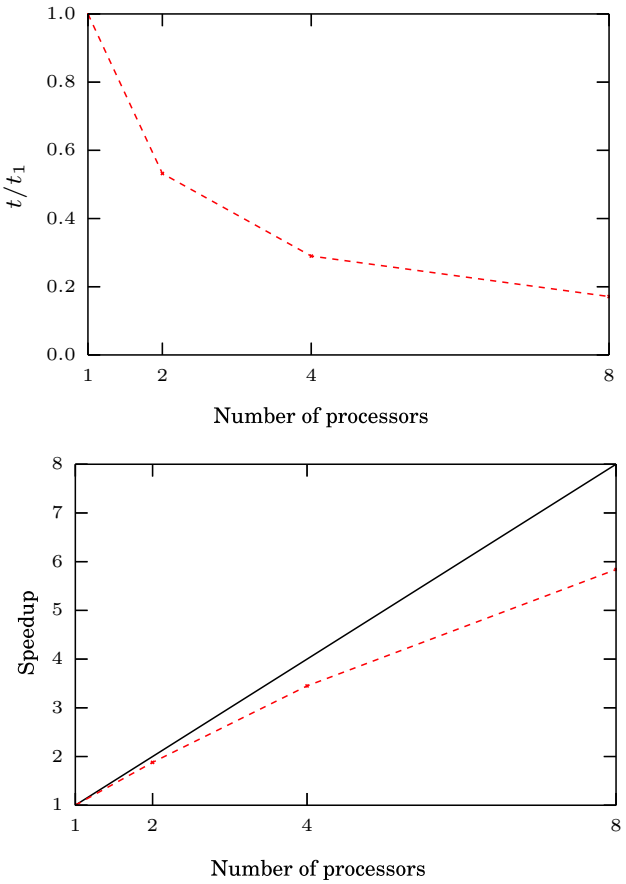


Fig. 29 Scalability test results

the stages introduced in Fig. 4 in Sect. 3 where the Local Resolution was presented split into a Fast Intersection Test (Sect. 4) and the Double Hierarchy Method (Sect. 5). Fig. 30 shows the results for the first 0.5 s of simulation which correspond to 1.5 turns of the helices. The cumulative counts for the following variables are represented:

- FE Potential: The number of times the Fast Intersection Test (Sect. 4) is called (number of FE potential neighbours to be checked) averaged to the number of particles.
- FE with contact: The average number of FE per particle that yields a positive result (have intersection with sphere) in the Fast Intersection Test.
- Entity with valid contact: The average number of relevant entities per particle determined by the H^2 Method.

Figure 30 shows that the number of Potential Neighbours to be treated is large compared to the FE with actual contact, a ratio of 30:1. Due to this fact and the improvement in performance shown in Table 10 it can be concluded that it is a good choice to perform the split which additionally brings modularity to the code.

7 Method limitations

7.1 Normal force in concave transitions

A limitation of this method which is common to the revised penalty-based contact algorithms occurs when a DE contacts with a slightly non-convex surface. Here the error introduced by the method is analysed and quantified for normal forces in the case of spherical DE in concave transitions.

The penalty method introduces an indentation which accounts for the local elastic deformation of the discrete element during a contact event and allows the imposition of the contact condition in a weak form. The use of rigid geometries with non-physical indentation introduces error in the contact detection. Constitutive laws such as Hertz-Mindlin present a limitation in terms of small deformation in order to work fine. This rule does not apply, however, for non-smooth regions where the basic assumptions are not met and contact detection errors arise.

A sphere moves horizontally in a plane π_a until it reaches a transition with other plane π_b which forms an acute angle α with the plane π_a (Fig. 31a). In this situation, a region can be defined between the current contact plane π_a and the plane π_n formed by the common edge and the normal of the second plane n_b. Whenever the sphere centre is in that region a discontinuity in forces will occur. The contact with plane π_b is detected only when the centre C has a normal projection onto the plane π_b forming a tangential contact. Figure 31b shows that when the new contact is detected, some indentation t is existing already and, therefore, the new contact force value introduces a discontinuity.

From the geometrical relations, the error ξ can be quantified as a ratio of the absolute value of the new force $\|\boldsymbol{F}_{n_b}\|$ over the absolute value of the current force $\|\boldsymbol{F}_{n_a}\|$. This value can be expressed in function of the change of angle α and indentation ratio t/δ relative to the sphere radius R:

$$\xi = \frac{\|\boldsymbol{F}_{n_b}\|}{\|\boldsymbol{F}_{n_a}\|} = \begin{cases} t/\delta & \text{for linear case} \\ (t/\delta)^{3/2} & \text{for Hertzian case} \end{cases} \tag{40}$$

Using the geometrical relationships and setting $\gamma = \delta/R$ as the relative indentation measure, the following expression is obtained:

$$t = \frac{R(\cos(\alpha) + \gamma - 1)}{\cos(\alpha)} \tag{41}$$

Finally the following expression is found:

$$\xi = \begin{cases} \frac{\cos(\alpha)+\gamma-1}{\gamma\cos(\alpha)} & \text{for linear case} \\ \left(\frac{\cos(\alpha)+\gamma-1}{\gamma\cos(\alpha)}\right)^{3/2} & \text{for Hertzian case} \end{cases} \tag{42}$$

The solution is plotted for the two cases (linear and Hertzian) for a different change of angle α and different γ indentation ratios.

Figure 32 shows that for an indentation of 1 % of the radius ($\gamma = 0.01$) and a small change in the angle of about 10° no error is produced, however, for an indentation of 3.3 % the error measure reaches a value of $\xi = 0.41$ for the Hertzian case ($\xi = 0.55$ for the linear case) which turns into a sudden

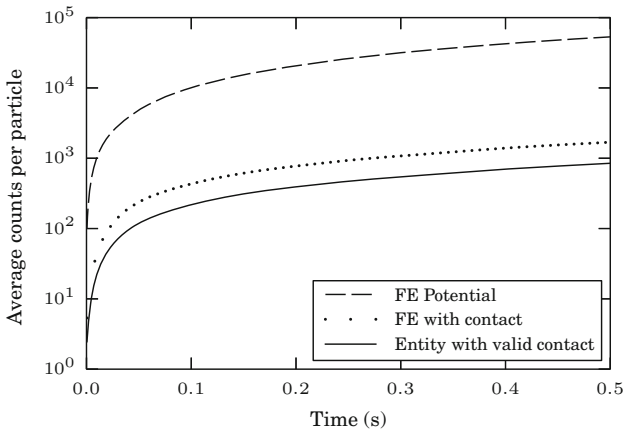


Fig. 30 Counts of FE checks in different stages

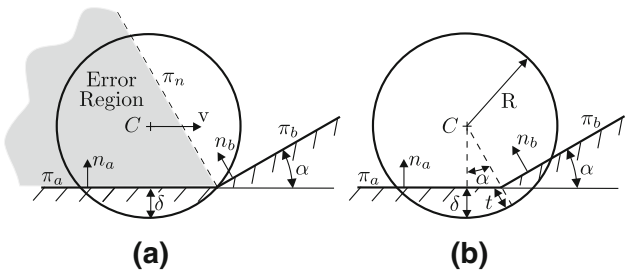


Fig. 31 Error emerging in concave transitions. **a** Error region. **b** Contact with 2 planes

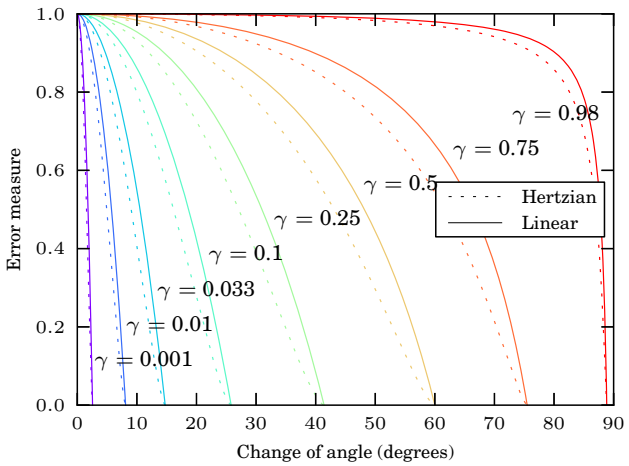


Fig. 32 Values of ξ measure error in function of change of angle α and indentation ratio γ

force of magnitude $\|\boldsymbol{F}_{n_b}\| = 0.41\|\boldsymbol{F}_{n_a}\|$ in the direction of n_b. The error tends to 0 as the angle change tends to 90° and does not occur for obtuse angles. On the other hand, the lower the change of angle α is, the greater the error is. It is bounded to 100 % of error $\xi = 1.0$ for the extreme case of coplanar transition. Luckily this very frequent case is considered by the Distance Hierarchy (Sect. 5.2) where a tolerance is used to detect the coplanar cases. Note that the error depends only on geometrical conditions and the indentation ratio relative to the sphere and not to the boundary FE mesh quality, the dependence of which has been solved using the Double Hierarchy Method.

7.2 Tangential force across elements

In many DEM codes, the tangential force is applied by means of an incremental scheme which requires to keep track of the forces that the particle has with each neighbour. The problem arises when a particle moves across two elements and the historical tangential force resets to zero because the contact is determined with the new element. This is common to all reviewed contact algorithms in the introduction. Even though it could be solved with a complex data structure, the error introduced is very low for practical situations. This has been proved in Sect. 6 where even the case of mesh independence, which combines tangential sliding and rolling, showed a negligible error.

The cases with larger error will be the ones with sliding where the tangential force is kept at its maximum (generally Coulomb friction value) and are much higher than during a rolling event. In this situation the error can be measured in terms of the missing work in a force-displacement diagram as the one shown in Fig. 33 which corresponds to a linear contact law [7,33] for normal and tangential directions.

In average, a particle with linear stiffness values K_n and K_t sliding across a transition of finite elements of characteristic length L with a relative indentation of δ will have the following error in the work done by the tangential force:

$$E_t = \frac{\|Error\ Area\|}{\|Total\ Area\|} = \frac{1/2(\mu\delta K_n)^2/K_t}{\mu\delta K_n L - 1/2(\mu\delta K_n)^2/K_t} = \frac{\mu\delta K_n/K_t}{2L - \mu\delta K_n/K_t} \approx \frac{\mu}{2}\frac{K_n}{K_t}\frac{\delta}{L} \tag{43}$$

For example, using the linear model in Shäfer [33] which suggests a ratio K_n/K_t of 7/2, with a particle-structure friction coefficient $\mu = 0.3$, the error in the integral of the tangential forces over the displacement has a value of $E_t = 0.525\,\delta/L$. That means that a large indentation of 10 % the characteristic size L of the FE yields only an error of approximately 5 %.

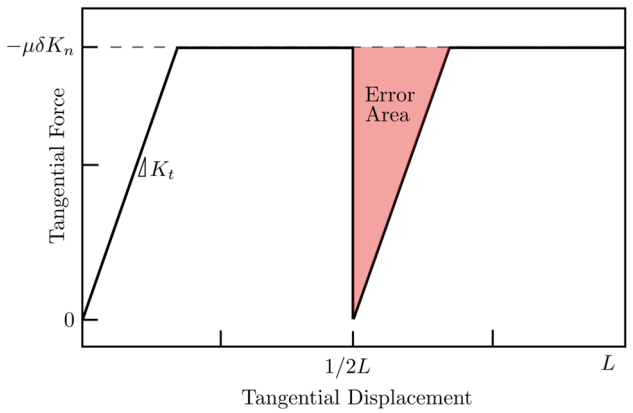


Fig. 33 Schematic force-displacement diagram with the discontinuity introduced by an element transition during a sliding event with a linear contact law

8 Concluding remarks

The new contact detection algorithm, the Double Hierarchy Method, is presented in this paper. The method, besides being accurate, robust and efficient, has been developed to perform well in extreme cases, where DEs and FEs sizes are different or where the relative indentation between them can be considerably high. This method can be used with different types of contact FEs providing a high level of accuracy in terms of contact force continuity in FE transitions and allowing multi-contact scenarios with high mesh independence and low effort. It has been designed to make it easy to implement and adapt to an existing DEM code. In addition, the algorithm has been conceived to be fully parallelizable, something essential in order to allow the calculation of real cases with a great amount of discrete and finite elements.

Contact calculation is split into two stages: Global Neighbour Search and Local Contact Resolution. Furthermore, the Local Contact Resolution level is split into two phases. The first one, Fast Intersection Test, aiming to determine which FE is in contact with each DE, discards very efficiently all the FEs not contacting the DE. Once the FEs with contact are known, the second phase, Double Hierarchy, takes place in order to accurately calculate the contact characteristics and to remove invalid contacts.

The accuracy and robustness of the proposed algorithm have been verified with different benchmark tests. Three tests were developed in order to evaluate contact elimination and energy conservation (see Fig. 18) in situations that involve the three possible *master contact entity*: a facet, an edge and a vertex. Additional tests show how the code behaves regarding to multi-contact (see Fig. 22a), contact continuity (see Fig. 20) and mesh independence (see Fig. 24).

An industrial example (Sect. 6.2) is presented to show the computational efficiency. Shared memory parallelization has a performance which is limited by the amount of serial parts

of the code. Having the possibility to parallelize an important part of the code, which is the contact detection, allows the computation speed to scale up. The results proved that the split of the Local Resolution into a Fast Intersection and the Double Hierarchy Method greatly improves the overall performance.

We highlight that there are some limitations for the use of this method that have been clearly defined and analysed in this paper. First, any geometry can be used for the contact elements (triangle, quadrilateral, polygon) only if they are convex as it has been detailed in Sect. 5.3. In second place it has been highlighted that this method is best suited when predicting the contact forces on the DE side and should be only used to calculate the forces on the FE side under certain conditions detailed in Sect. 2.2.2. Finally it has been appointed that the method works perfectly in for the normal force calculation with surfaces that present a convex transition even for very large indentations (see examples in Sect. 6.1). It may present some discontinuities in the normal contact forces, however, when dealing with concave transitions between surfaces that form small angles if the indentation is not small. Also, in sliding situations across different elements which a non-small indentations, some error in the continuity of tangential contact forces can arise. The error introduced in these cases, which is most of the times negligible for practical cases, has been quantified in Sect. 7.

Notwithstanding those limitations, the proposed algorithm can be used in a wide range of DE/FE simulations.

Acknowledgments This work has been carried out with the financial support Spanish MINECO within the BALAMED project (BIA2012-39172).

References

1. Andrade JE, Lim KW, Avila CF, Vlahinić I (2012) Granular element method for computational particle mechanics. Comput Methods Appl 241–244:262–274
2. Bonet J, Peraire J (1991) An alternating digital tree (adt) algorithm for 3d geometric searching and intersection problems. Int J Numer Methods Eng 31(1):1–17
3. Boon CW, Houlsby GT, Utili S (2012) A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method. Comput Geotech 44:73–82
4. Boon CW, Houlsby GT, Utili S (2013) A new contact detection algorithm for three-dimensional non-spherical particles. Powder Technol 248:94–102
5. Casas G, Mukherjee D, Celigueta MA, Zohdi TI, Onate E (2015) A modular, partitioned, discrete element framework for industrial grain distribution systems with rotating machinery. Comput Part Mech. doi:10.1007/s40571-015-0089-9
6. Chen H, Zhang YX, Zang M, Hazell PJ (2015) An accurate and robust contact detection algorithm for particle-solid interaction in combined finite-discrete element analysis. Int J Numer Methods Eng 103(8):598–624
7. Cundall P, Strack O (1979) A discrete numerical model for granular assemblies. Géotechnique 29(1):47–65
8. Dadvand P, Rossi R, Oñate E (2010) An object-oriented environment for developing finite element codes for multi-disciplinary applications. Arch Comput Methods Eng 17(3):253–297
9. Dang HK, Meguid MA (2013) An efficient finite-discrete element method for quasi-static nonlinear soil-structure interaction problems. Int J Numer Anal Methods Geomech 37(2):130–149
10. Eliáš J (2014) Simulation of railway ballast using crushable polyhedral particles. Powder Technol 264:458–465
11. Ericson C (2004) Real-time collision detection (The Morgan Kaufmann series in interactive 3D technology). Morgan Kaufmann Publishers Inc., San Francisco
12. Feng YT, Owen DRJ (2002) An augmented spatial digital tree algorithm for contact detection in computational mechanics. Int J Numer Methods Eng 55(2):159–176
13. Garcia X, Xiang J, Latham JP, Harrison JP (2009) A clustered overlapping sphere algorithm to represent real particles in discrete element modelling. Géotechnique 59(9):779–784
14. Han K, Peric D, Crook A, Owen D (2000a) A combined finite/discrete element simulation of shot peening processes—Part I: studies on 2D interaction laws. Eng Comput 17(5):593–620
15. Han K, Peric D, Owen D, Yu J (2000b) A combined finite/discrete element simulation of shot peening processes—part II: 3D interaction laws. Eng Comput 17(6):680–702
16. Han K, Feng Y, Owen D (2007) Performance comparisons of tree-based and cell-based contact detection algorithms. Eng Comput 24(2):165–181
17. Horner DA, Peters JF, Carrillo A (2001) Large scale discrete element modeling of vehicle-soil interaction. J Eng Mech-ASCE 127(10):1027–1032
18. Hu L, Hu G, Fang Z, Zhang Y (2013) A new algorithm for contact detection between spherical particle and triangulated mesh boundary in discrete element method simulations. Int J Numer Methods Eng 94(8):787–804
19. Irazábal González J (2015) Numerical modeling of railway ballast using the discrete element method. Master thesis
20. Karabassi EA, Papaioannou G, Theoharis T, Boehm A (1999) Intersection test for collision detection in particle systems. J Graph Tools 4(1):25–37
21. Kodam M, Bharadwaj R, Curtis J, Hancock B, Wassgren C (2009) Force model considerations for glued-sphere discrete element method simulations. Chem Eng Sci 64(15):3466–3475
22. Kremmer M, Favier JF (2001) A method for representing boundaries in discrete element modelling—part II: kinematics. Int J Numer Methods Eng 51(12):1423–1436
23. Meyer M, Barr A, Lee H, Desbrun M (2002) Generalized barycentric coordinates on irregular polygons. J Graph Tools 7(1):13–22
24. Michael M, Vogel F, Peters B (2015) DEM-FEM coupling simulations of the interactions between a tire tread and granular terrain. Comput Methods Appl Mech Eng 289:227–248
25. Mindlin R (1949) Compliance of elastic bodies in contact. J Appl Mech 16:259–268
26. Munjiza A (2004) The combined finite-discrete element method, 1st edn. Wiley, Hoboken
27. Munjiza A, Andrews KRF (1998) NBS contact detection algorithm for bodies of similar size. Int J Numer Methods Eng 43(1):131–149
28. Nakashima H, Oida A (2004) Algorithm and implementation of soil–tire contact analysis code based on dynamic FE–DE method. J Terramech 41(2–3):127–137, 14th International Conference of the ISTVS
29. Nezami EG, Hashash YMA, Zhao D, Ghaboussi J (2004) A fast contact detection algorithm for 3-D discrete element method. Comput Geotech 31(7):575–587
30. Oñate E, Rojek J (2004) Combination of discrete element and finite element method for analysis of geomechanics problems. Comput Methods Appl Mech Eng 193:3087–3128

31. Oñate E, Zárate F, Miquel J, Santasusana M, Celigueta MA, Arrufat F, Gandikota R, Valiullin K, Ring L (2015) A local constitutive model for the discrete element method. Application to geomaterials and concrete. Comput Part Mech 2(2):139–160
32. Renzo AD, Maio FPD (2004) Comparison of contact-force models for the simulation of collisions in dem-based granular flow codes. Chem Eng Sci 59(3):525–541
33. Shäfer J, Dippel S, Wolf D (1996) Force schemes in simulation of granular materials. Journal de Physique I 6(1):5–22
34. Su J, Gu Z, Xu XY (2011) Discrete element simulation of particle flow in arbitrarily complex geometries. Chem Eng Sci 66(23):6069–6088
35. Su J, Gu Z, Zhang M, Xu XY (2014) An improved version of RIGID for discrete element simulation of particle flows with arbitrarily complex geometries. Powder Technol 253:393–405
36. Sukumar N, Tabarraei A (2004) Conforming polygonal finite elements. Int J Numer Methods Eng 61(12):2045–2066
37. Thornton C, Cummins SJ, Cleary PW (2011) An investigation of the comparative behaviour of alternative contact force models during elastic collisions. Powder Technol 210(3):189–197
38. Wang SP, Nakamachi E (1997) The inside-outside contact search algorithm for finite element analysis. Int J Numer Methods Eng 40(19):3665–3685
39. Wellmann C (2011) A two-scale model of granular materials using a coupled DE-FE appoach. Ph.D. thesis, Institut für Kontinuumsmechanik, Leibniz Universität Hannover
40. Wensrich CM, Katterfeld A (2012) Rolling friction as a technique for modelling particle shape in DEM. Powder Technol 217:409–417
41. Williams JR, O'Connor R (1999) Discrete element simulation and the contact problem. Arch Comput Methods Eng 6(4):279–304
42. Williams JR, Perkins E, Cook B (2004) A contact algorithm for partitioning N arbitrary sized objects. Eng Comput 21(2/3/4):235–248
43. Wriggers P (2002) Computational contact mechanics. Wiley, Chichester
44. Zang M, Gao W, Lei Z (2011) A contact algorithm for 3D discrete and finite element contact problems based on penalty function method. Comput Mech 48:541–550
45. Zhong Z (1993) Finite element procedures for contact-impact problems. Oxford University Press, New York
46. Zienkiewicz O, Taylor R, Fox D (2014) The finite element method for solid and structural mechanics, 7th edn. Butterworth-Heinemann, Oxford